Data Driven Methods for Large Scale Network Level Traffic Modeling

Rezaur Rahman
University of Central Florida
DATA-DRIVEN METHODS FOR LARGE SCALE NETWORK LEVEL TRAFFIC MODELING

by

REZAUR RAHMAN

B.Sc. Bangladesh University of Engineering and Technology, 2015
M.Sc. University of Central Florida, 2019

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Civil, Environmental, and Construction Engineering in the College of Engineering and Computer Science at the University of Central Florida Orlando, Florida

Fall Term 2021

Major Professor: Samiul Hasan
ABSTRACT

Rapid growth in population along with urban-centric activities impose a massive demand on existing transportation systems, thus increasing traffic congestion and other mobility related challenges. To overcome such challenges, we need network-scale models to accurately predict real-time traffic demand and associated congestion. However, traditional network modeling approaches have shortcomings due to the complexity in traffic flow modeling, limited scope to incorporate real-time data available from emerging data sources and requiring excessive computation time to generate accurate estimation of traffic flows. Advancement in traffic sensing technologies with big data has created a new opportunity to overcome these challenges and implement deployable data-driven models to predict network-level traffic dynamics and congestion propagation in real time. However, existing data-driven approaches are limited in scope: they are developed for small-scale networks; they do not consider the fundamental concept of traffic flow propagation; and they are applied for short-term prediction (<1 hour). In this dissertation, we develop graph convolution based neural network architectures for network scale traffic modeling as a solution to overcome these limitations. First, we develop a Graph Convolutional Neural Network (GCNN) Model to solve the traffic assignment problem in a data-driven way; the validation results show that the model can learn the user equilibrium traffic flow well (mean error <2%). Since the model can instantaneously determine the traffic flows of a large-scale network, this approach can overcome the challenges of deploying mathematical programming or simulation-based traffic assignment solutions for large-scale networks. Second, we scale this approach and develop a Graph Convolutional LSTM (GCN-LSTM) model for traffic movement volume prediction at intersection level. We rigorously tested the model over traffic movement volume data collected from Seminole County’s automated signal performance measure
(ATSPM) database which show that 90% of cases, absolute error of the predicted values is less than 20. Finally, we develop a Dynamic Graph Convolutional LSTM (DGCN-LSTM) model to predict evacuation traffic flow for interstate network of Florida. The implemented model can be applied to predict evacuation traffic over a longer forecasting horizon (6-hour) with higher accuracy ($R^2$ score 0.95). Hence, it can assist transportation agencies to activate appropriate traffic management strategies to reduce delays for evacuating traffic.
ACKNOWLEDGMENTS

I would like to convey my heartiest gratitude to my honorable supervisor Dr. Samiul Hasan for his continuous support and supervision throughout my PhD journey. Your expertise was invaluable in formulating the research questions and methodology. On the top of that, your insightful feedbacks pushed me to refine my knowledge and enhance my research to a higher level.

In addition, I would like to acknowledge my lab mates for their assistance, questions, and insights, which helped me to consolidate my research ideas. I would particularly like to thank Dr. Kamol Roy for helping me to find the right tools to accelerate my research experiments.

I would also like to thank my wife Farzana Rahman for being considerate and supportive, I could not have completed this dissertation on time without your constant support. Finally, I would like to thank my family members and friends for their wise advice, encouragement and compassionate ear. You all gave me inspiring discussions as well as happy distractions to rest my mind outside of my research.
# TABLE OF CONTENT

LIST OF FIGURES ........................................................................................................................................... x
LIST OF TABLES ............................................................................................................................................... xiii
CHAPTER 1: INTRODUCTION ............................................................................................................................ 1
  1.1 Background .................................................................................................................................................. 1
  1.2 Motivation .................................................................................................................................................. 2
  1.3 Dissertation Objectives ............................................................................................................................. 5
  1.4 Contributions ............................................................................................................................................ 6
  1.5 Structure of the Dissertation ..................................................................................................................... 7

CHAPTER 2: DATA-DRIVEN TRAFFIC ASSIGNMENT: A NOVEL APPROACH OF LEARNING TRAFFIC FLOW PATTERNS ........................................................................................................... 9
  2.1 Introduction ................................................................................................................................................ 9
  2.2 Data-driven Traffic Assignment ................................................................................................................ 10
    2.2.1 Problem Definition ............................................................................................................................... 10
    2.2.2 Graph Convolution Neural Network for Flow Pattern Learning ...................................................... 14
    2.2.3 Modeling Traffic Flow using Diffusion Graph Convolution .............................................................. 17
  2.3 Data Generation ........................................................................................................................................ 20
  2.4 Results and Discussion ............................................................................................................................... 23
    2.4.1 Result interpretation .............................................................................................................................. 27
    2.4.2 Stability of the Solution ....................................................................................................................... 32
  2.5 Conclusions ................................................................................................................................................. 35

CHAPTER 3: A DATA-DRIVEN NETWORK MODEL FOR TRAFFIC VOLUME PREDICTION AT SIGNALIZED INTERSECTIONS ............................................................................................................... 36
  3.1 Introduction ............................................................................................................................................... 36
  3.2 Literature Review ..................................................................................................................................... 37
3.3 Problem Formulation ........................................................................................................ 40
3.4 Methodology .................................................................................................................... 43
    3.4.1 Graph Convolution for Spatial Dependency Modeling ........................................... 43
    3.4.2 LSTM Model for Temporal Dependency Modeling .................................................. 44
3.5 Modeling Frameworks ..................................................................................................... 45
    3.5.1 Graph Convolutional LSTM ................................................................................... 45
    3.5.2 Graph Convolutional Encoder Decoder LSTM ....................................................... 47
3.6 Data Collection and Preprocessing .................................................................................. 48
    3.6.1 Traffic Movement Volumes ..................................................................................... 48
    3.6.2 Travel Demand Features ....................................................................................... 52
3.7 Experiments .................................................................................................................... 54
    3.7.1 Input features and Graph Representation .................................................................... 54
    3.7.2 Baseline Models ...................................................................................................... 56
    3.7.3 Model Training ....................................................................................................... 56
    3.7.4 Model Training ....................................................................................................... 58
3.8 Conclusions ..................................................................................................................... 65

CHAPTER 4: UNDERSTANING NETWORK WIDE HURRICANE EVACUATION TRAFFIC PATTERN FROM LARGE-SCALE TRAFFIC DETECTOR DATA .......................... 66
4.1 Introduction ...................................................................................................................... 66
4.2 Data Description .............................................................................................................. 67
4.3 Data Preprocessing .......................................................................................................... 69
4.4 Data Exploration .............................................................................................................. 73
    4.4.1 Spatiotemporal Pattern of Population Under Evacuation ....................................... 73
    4.4.2 Spatiotemporal Patterns of Evacuation Traffic ....................................................... 75
4.5 Results .......................................................................................................................... 78
  4.5.1 Linear Regression ................................................................................................. 78
  4.5.2 Tree Based Model to Estimate the Feature Importance ...................................... 81
  4.6 Conclusions ............................................................................................................... 82

CHAPTER 5: A DEEP LEARNING APPROACH FOR NETWORK-WIDE DYNAMIC
TRAFFIC PREDICTION DURING HURRICANE EVACUATION ........................................ 84
  5.1 Introduction ............................................................................................................... 84
  5.2 Literature Review ..................................................................................................... 85
  5.3 Problem Formulation .............................................................................................. 87
  5.4 Methodology ............................................................................................................ 90
    5.4.1 Learning Traffic Flow Dynamics of the Transportation Network ....................... 90
    5.4.2 Network-wide Evacuation Traffic Prediction ..................................................... 92
  5.5 Data Collection and Preprocessing .......................................................................... 94
    5.5.1 Traffic Detector Data ........................................................................................ 94
    5.5.2 Zonal Level Mandatory Evacuation .................................................................. 96
  5.6 Experiments ............................................................................................................. 98
    5.6.1 Feature Extraction and Graph Representation .................................................... 98
    5.6.2 Baseline Models ................................................................................................ 102
    5.6.3 Model Training .................................................................................................. 103
  5.7 Experiment results ................................................................................................... 105
  5.8 Congestion Mapping to Understand Network Disruption ..................................... 108
  5.9 Conclusions ............................................................................................................. 109

CHAPTER 6: CONCLUSIONS ............................................................................................ 111
  6.1 Summary of Major Results ..................................................................................... 111
6.2 Limitations and Future Research Directions

REFERENCES
LIST OF FIGURES

Figure 2.1 Data-Driven traffic assignment problem................................. 12
Figure 2.2 GCNN architecture for traffic flow pattern learning..................... 15
Figure 2.3 Transportation networks (a) Sioux Falls network [38] (b) East Massachusetts network (US Census Bureau, 2015) [39] ......................................................... 21
Figure 2.4 Illustrates the distribution of flow-capacity ratio for different traffic conditions ................................................................................................................................. 23
Figure 2.5 Comparison between actual and estimated link flows for a given OD demand (Sioux Falls Network) ......................................................................................................................... 26
Figure 2.6 Comparison between actual and estimated link flows for a given OD demand (East Massachusetts Network) ......................................................................................................................... 27
Figure 2.7 Variations of Betweenness Centrality for different nodes at different traffic condition (Sioux Falls Network) ......................................................................................................................... 29
Figure 2.8 Distribution of weight ($w_q$) over different traffic conditions ............ 31
Figure 2.9 Loss function values for Laplacian spectral Graph .......................... 34
Figure 3.1 Model architecture (a) Graph convolutional neural network (b) LSTM neural network ................................................................................................................................. 43
Figure 3.2 Modeling framework (a) Graph Convolutional LSTM (b) Graph Convolutional Encoder Decoder LSTM ......................................................................................................................... 47
Figure 3.3 The study area (Open Street Map) .................................................. 49
Figure 3.4 Data processing framework ................................................................ 50
Figure 3.5 Distribution of data samples based on vehicles per hour per lane ........ 52
Figure 3.6 Traffic analysis zones (TAZs) and Sub zones for the analysis ............ 53
Figure 3.7 Variations of training and validation loss values GCNLSTM model................. 58

Figure 3.8 Performance of the model in turning and through movement prediction ....... 61

Figure 3.9 Predicted through movement volumes for all intersections for a given weekday (Wednesday) time period (timestamp 08-10-2016 14:00:00 - 18:00:00) ........................................ 63

Figure 3.10 Predicted through movement volumes for all intersections for a given weekend (Sunday) time period (timestamp 2016-11-13 14:00:00-18:00:00) ........................................ 64

Figure 4.1 Distributions of the 1426 detectors in the study network ............................. 69

Figure 4.2 Framework for data processing ..................................................................... 70

Figure 4.3 Distribution of data samples based on vehicles per hour per lane.................. 72

Figure 4.4 (a) Zone wise mandatory evacuation orders with corresponding declaration dates (b) Population of mandatory evacuation zones ......................................................... 74

Figure 4.5 Spatiotemporal traffic flow distribution, here we plot the traffic detectors based on (latitudes, longitudes)/unique zone ids ........................................................................ 77

Figure 4.6 Spatial traffic flow pattern during hurricane Irma’s evacuation period over different days (Sep. 06- Sep. 09, 2017) ............................................................................................ 78

Figure 4.7 Illustrates (a) features importance (b) Root Mean Square Error (RMSE) and (c) R² Score for the tree-based models ..................................................................................... 81

Figure 5.1 A dynamic graph learning approach for network-wide traffic prediction ....... 91

Figure 5.2 A deep learning architecture for evacuation traffic prediction ...................... 93

Figure 5.3 Network of inter-state highways .................................................................... 95

Figure 5.4 Framework for data processing ...................................................................... 96

Figure 5.5 Temporal variations of total population under mandatory evacuation ......... 98
Figure 5.6 Variations of training and validation loss (a) DGCN-LSTM (b) Transfer Learned DGCN-LSTM .................................................................................................................................................. 104

Figure 5.7 Correlation between actual and predicted values for different prediction horizons .................................................................................................................................................. 107

Figure 5.8 Actual and predicted value for different detectors for 6 hour prediction horizon .................................................................................................................................................. 107

Figure 5.9 Congestion map indicating traffic flow (predicted) variations over different time during hurricane evacuation .................................................................................................................................................. 109
LIST OF TABLES

Table 2.1  List of notations ................................................................. 13
Table 2.2  Model performances based on test dataset .......................... 25
Table 3.1  Description of the notations associated with the model development .......... 42
Table 3.2  Description of input features .................................................. 55
Table 3.3  Comparisons among different models to predict traffic volume over 4-hour sequence for different movement types ........................................................................ 59
Table 4.1  Estimates from linear regression model.................................. 79
Table 5.1  Description of the notation associated with the model development ............ 89
Table 5.2  Description of the input features ............................................... 100
Table 5.3  Comparisons among different models to predict traffic over 6-hour sequence 106
CHAPTER 1: INTRODUCTION

1.1 Background

Over the past decades rapid population growth with increasing urban-centric activities has imposed a massive demand on urban transportation systems. Consequently, urban regions are faced with increasing traffic congestion and other mobility-related problems such as traffic crashes, excessive fuel consumption; especially in major cities where the level of human activities is high. According to 2018 Global Traffic Scorecard Americans lost an average of 97 hours a year due to congestion, costing them nearly $87 billion, an average of $1,348 per driver [1]. Roadway capacity constraints, traffic incidents and mismanagement of traffic signal timing at intersections are some of the major reasons behind traffic congestion. To overcome these challenges, we need to optimize the use of existing road network, ensure proactive traffic control and management, which largely depends on how accurately we can capture traffic flow patterns to predict traffic for entire roadway network.

However, network level traffic modeling is a complex phenomenon that involves modeling traffic dynamics and replicating the prevailing traffic conditions using the interactions between the demand and supply models. Traditionally, network modeling approaches (e.g., static or dynamic traffic assignment) rely on mathematical formulation and simulation-based approaches to estimate traffic flows. These approaches rely on several assumptions of user behavior and knowledge such as: (i) drivers have precise information and knowledge about the underlying network; (ii) drivers make rational choices when choosing a route; and (iii) all drivers are homogeneous (Kim et al., 2009). Although some of these assumptions may not hold in a real-world scenario, this approach provides the most reasonable solutions of the traffic assignment problem. Nevertheless, the complexity and computation time required to reach reasonable solution, make these methods unsuitable for real-time deployment. Researchers are exploring to develop efficient solution
methods for traffic estimation and prediction which can be deployed for real-time transportation operation purposes.

Emerging sensing technologies such as mobile phones, social media (Twitter), subway smart card transactions and taxicab GPS observations along with traditional traffic sensors such as loop detectors, microwave vehicle detection, bluetooth detector etc. have created a new opportunity for us to deal with such challenges using big data. Recent trends in transportation research show that, researchers are more focused on developing data-driven methods to capture complex non-linearities in traffic dynamics. One of the major benefits of data-driven methods is that they are easy to deploy in a real-time context, which will eventually help us making proactive decisions for efficient traffic management.

This dissertation adopts a data-driven approach towards developing new methodologies to model network-level traffic patterns and predict traffic flow variations using real-time data obtained from traffic sensors. Real-time network level traffic modeling will help traffic management agencies understand the prevailing and future traffic conditions for each link of the network. Consequently, they will be able to make proactive decisions for traffic optimization and effective traffic management, especially in case of emergency event such as hurricane evacuation.

1.2 Motivation

Network wide traffic modeling has been an integral part of proactive traffic management system. Researchers have established a wide variety of approaches to capture the traffic behavior for large scale network. For example, static traffic assignment (STA) method has widely been used for solving traffic assignment problem for flow estimation. This method has been quite successful in network modeling for transportation planning purpose. However, static traffic assignment solution approach has two major limitations [2]. First, it relies on certain assumptions for users’ route
choice behavior which depends on various interacting factors (e.g. travelers’ knowledge of prevailing traffic condition), including some unobserved heterogeneity (e.g. travelers always make rational choice), in some cases the behavioral assumptions fail to represent the real-world scenario. Second, it assumes constant Origin-Destination (OD) demand whereas traffic demand may significantly change over time, which makes it impractical to deploy for real-world traffic scenario.

As an alternative to STA, the dynamic traffic assignment (DTA) [3, 4, 13–20, 5–12] method captures complex traffic dynamics in a realistic way. It replicates traffic phenomena using the interactions between demand variations and supply models. Nevertheless, the state-of-the-art DTA methods also rely on similar behavioral assumptions as the STA methods to formulate the modeling framework. DTA model aims to describe the time-varying network and demand interaction more precisely which makes it more complex, thus requires more computational time. In a smaller model with less complexity, the DTA process works well in converging to a point of equilibrium. As the scale of a model grows in size, complexity and congestion, the DTA process becomes more difficult and time-consuming, rendering it unrealistic for real-time deployment.

Over the past decades, advancement in traffic sensing technologies such as roadway traffic detectors, GPS devices etc. has provided us with information on real-time traffic conditions for a transportation network. However, traditional network modeling approaches are limited in scope and unable to utilize all these data sources while predicting future traffic. Hence, these models are less robust, which means model developed for a specific traffic condition (e.g. lower demand condition) will perform poorly to a different traffic condition (e.g. higher demand condition) in absence of significant calibration process. In addition, calibration process of such models is
complex, and time consuming, hence they are less suitable for real-world deployment, such as real-time traffic prediction.

Such limitations have encouraged us to look for alternative data-driven approaches for network scale traffic modeling. Unlike traditional models, data driven network models can capture user route choice behavior extracting information from traffic detectors and passive data sources rather than relying on unrealistic assumptions. Moreover, since data driven approaches rely on real time traffic variations to predict future traffic, they are more robust and demand responsive. Hence, data driven network models can be applied for real-time traffic prediction, which is one critical aspect of managing traffic incident and emergency events such as hurricane evacuation. However, one of the major challenges in developing data-driven network model is to implement a framework that can learn the relationship between OD demand and link flows from real-world data.

In recent years, researchers are exploring different data-driven approaches for traffic modeling and prediction. However, existing data-driven approaches have several limitations such as, they predict only short-term traffic states (speed, flow, travel time) (Billings and Jiann-Shiou, 2006; Deshpande and Bajaj, 2016; Lee, 2009; Vlahogianni et al., 2014; Wu et al., 2004; Yu et al., 2016) for one or multiple segments of highways, but not at the scale of a network (Gu et al., 2019; Rahman and Hasan, 2018; Song et al., 2018). Moreover, these approaches do not consider features related to travel demand and land use when predicting future traffic. As such, these approaches consider traffic prediction as a simple time series problem and predict the traffic state for a shorter time horizon (e.g., next 5 to 15 min). However, network level traffic prediction is more challenging due to the higher computational complexity required by the network topology. Such a problem requires (i) capturing the spatial cross correlation of traffic demand related features in a roadway
network, (Polson and Sokolov, 2017) and (ii) reflecting drivers’ route choice and associated congestion propagation inside the network.

Recently, graph theory coupled with generalized neural network architecture has been utilized to model the dynamics between the structural properties and the functions of networks (Atwood and Towsley, 2015; Cui et al., 2019; Li et al., 2018; Zhou et al., 2018) solving problems such as modeling physical systems, learning molecular fingerprints, predicting protein interface, and classifying diseases, which require that a model learns from graph-based inputs (Zhou et al., 2018). However, the application of such neural network architecture hardly exists for a large-scale transportation network. A few studies have proposed spatial and temporal learning-based methods for small scale traffic prediction (Cui et al., 2019; Li et al., 2018). These methods do not solve the fundamental traffic assignment problem considering flow propagation inside the network. Furthermore, these methods have not been tested in any realistic networks. Thus, in this study, first we will develop data-driven network modeling approaches to solve the traffic assignment problem. Second, we will test the scalability and transferability of such models for real-world traffic prediction application for large scale networks.

1.3 Dissertation Objectives

This dissertation presents novel data-driven methods for network level traffic modeling, utilizing emerging data sources and deep learning techniques to overcome the limitations of traditional mathematical modeling and simulation-based approaches. It focuses on the following specific objectives:

i. Develop a data-driven network scale model to solve the traffic assignment problem considering the relation between demand variations and traffic flow patterns of a transportation network.
ii. Develop a data-driven network model for intersection-level traffic movement volume prediction by learning spatiotemporal cross-correlation among zonal level traffic demand (Central Florida Regional Planning Model) related features.

iii. Develop transfer learning based deep learning architecture for network-wide dynamic traffic prediction over multiple time steps.

1.4 Contributions

This dissertation has made following contributions:

i. This dissertation offers a data-driven formulation of the traffic assignment problem which is completely different than previously used approaches and does not rely on assumptions related to user behavior. It develops a novel deep learning architecture to solve this traffic assignment problem which is more robust and deployable to real-world application.

ii. In addition, it shows the experimental evidence that a generalized neural network architecture can learn the user equilibrium traffic flow assignment of a transportation network without having any information on user behavior. These findings will open up new research direction in data-driven solution of network modeling problems.

iii. This research provides further empirical evidence that such data-driven methods are more robust and scalable to predict intersection level high-fidelity traffic for a large-scale network. Thus, it presents a predictive modeling-based analytics tool for proactive traffic management rather than relying on computationally expensive dynamic traffic assignment methods.

iv. This research develops a robust data-driven model for zonal level evacuation traffic prediction for long term period (>1-hour). The model overcomes the limitation of existing data-driven traffic prediction models to account for irregular traffic flow variations during
evacuation period. Such predictive model will assist emergency manager to take proactive strategies for efficient evacuation traffic management.

1.5 Structure of the Dissertation

In Chapter 2, we present a novel data-driven approach of learning traffic flow patterns of a transportation network given that many instances of OD demand and traffic flow of the network are available. Instead of estimating traffic flow patterns assuming certain user behavior (e.g., user equilibrium), here we explore the idea of learning those patterns from large-scale training data by developing a neural network architecture. In this work, we seek to answer the question: Can a deep learning model learn the flow patterns of a network without relying on the assumptions of user behavior for assigning traffic in the network? To answer this question, we develop a graph convolution based deep neural network and test the model performance to learn user equilibrium flow for two different networks: Sioux Falls and East Massachusetts.

In Chapter 3, we develop a new modeling framework for intersection--level movement volume prediction for a large-scale network. We develop a graph convolution based deep learning model to predict traffic movement volume by capturing the correlation between demand features and traffic flow patterns of a transportation network. To test the model, we have collected Automated Signal Performance Measure (ATSPM) data and zonal-level travel demand (CFRPM) data for Seminole County.

In Chapter 4, we build a data pipeline incorporating automated data quality checking, data imputation, and spatiotemporal visualization of network wide traffic to understand the changes in traffic patterns during hurricane evacuation. We have collected Microwave Vehicle Detection System (MVDS) data for four major highways in Florida: I-75, I-95, I-4 and Florida Turnpike, these routes serve most of the evacuation traffic during Hurricane Irma and extensively analyze
these data to understand the nature and extent of Hurricane Irma’s evacuation traffic and the quality of these data for evacuation traffic modeling. Moreover, different factors such as time and place of evacuation order, population under evacuation order, location of the evacuees, evacuation start time, and traffic congestion are critical to predict evacuation traffic demand in future hurricanes. Hence, we also perform an empirical analysis to understand the impact of these spatiotemporal factors on evacuation traffic patterns.

In Chapter 5, we develop a novel a dynamic graph convolutional LSTM (DGCN-LSTM) model for spatiotemporal traffic prediction during evacuation period. However, modeling spatiotemporal traffic flow patterns requires extensive data over a longer time period, whereas evacuations typically last for 2 to 5 days. We overcome this issue by adopting a transfer learning approach. We train the model for non-evacuation period traffic data and transfer the pre trained model for evacuation traffic prediction. We also included additional neural network-based layers to control the information flow from the pre-trained DGCN-LSTM model. The final transfer learned DGCN-LSTM model performs well to predict evacuation traffic flow.

Finally, in Chapter 6, we conclude the dissertation by discussing our overall findings, citing the limitations, and providing directions for future studies.
CHAPTER 2: DATA-DRIVEN TRAFFIC ASSIGNMENT: A NOVEL APPROACH OF LEARNING TRAFFIC FLOW PATTERNS

2.1 Introduction

In this chapter, we present a novel data-driven approach of learning traffic flow patterns of a transportation network given that many instances of origin-destination (OD) travel demand and traffic flow of the network are available. Instead of estimating traffic flow patterns assuming certain user behavior (e.g., user equilibrium), here we explore the idea of learning those patterns from large-scale training data by developing a neural network architecture. In particular, we use the Graph Convolutional Neural Network (GCNN) which generalize traditional neural networks to work on structured graphs [26]. GCNN utilize an adjacency matrix or a Laplacian matrix to represent the structure of a graph.

Recently, several studies have utilized the concept of graph convolution (GCN) to represent traffic network as a generalized graph for traffic state prediction [27, 28]. Moreover, Graph Convolution Neural Network (GCNN) has been emerging as a new approach to overcome the challenge of high dimensionality when predicting travel demand for a large-scale network [29, 30]. However, this is the first time, we structured a GCNN model to learn the traffic flow pattern of a transportation network through learning the diffusion process of multiple OD demands from nodes to links. In this work, we seek to answer the question: Can a deep learning model learn the flow patterns of a network without relying on the assumptions of user behavior for assigning traffic in the network? If positive, apart from assigning traffic in a purely data-driven way, this can open up new directions in transportation networks research offering efficient computational approaches.

---

for solving dynamic traffic assignment, network design, and many other problems with real-world applications. Thus, this study makes several contributions:

- It formulates the traffic-assignment problem as a data-driven learning problem considering the underlying transportation network and available instances of OD demand and link flows, without assuming any user behavior.
- It develops a novel neural network architecture that solves the learning problem to determine link flows based on OD demand and traffic flow data.
- It provides rigorously tested experimental evidence that such a neural network architecture can learn the user equilibrium traffic flow assignment of a transportation network only from data.

2.2 Data-driven Traffic Assignment

In this section, first we formulate the data-driven traffic assignment problem. We then describe the Graph Convolutional Neural Network (GCNN) approach to model traffic flow propagation in the network.

2.2.1 Problem Definition

In a transportation network, all nodes are connected, and each link is associated with information such as distance, speed limit, capacity etc. Here, we consider the transportation network as a weighted directed graph $G(v, E, A_w)$ where $v$ denotes the set of nodes and $E$ denotes the set of links between nodes $(i,j)$. $A_w$ represents the connectivity between nodes as a weighted adjacency matrix, where weights are based on free flow travel time between any two nodes $(i,j)$, defined as follows:

$$
A_w(i, j) = \begin{cases} 
 t_{i,j}^0 & \text{if } i \rightarrow j \\
 t_{j,i}^0 & \text{if } j \rightarrow i \\
 0, & \text{if } i = j 
\end{cases} 
$$

(2.1)
where, $t^0$ denotes the free flow travel time between the origin and the destination nodes. The proposed data-driven formulation of the traffic assignment problem aims to learn the flow patterns of a transportation network based on the network structure and the instances available on origin to destination (OD) travel demand and link flows (Figure 2.1). Also, we have the information on network characteristics, such as location of each node with respect to other nodes, travel distance or free flow travel time between different nodes. From this information, we develop a data-driven method to estimate the link flows for given travel demands. During the estimation process, we also learn the traffic flow propagation from origin nodes toward destination nodes in a transportation network. Let, $X$ be the demand matrix for the transportation network $G$, where each element of a row indicates the travel demand between origin node ($i$) and the destination node ($j$).

The traffic assignment problem aims to learn a function $\mathcal{F}(. )$ that maps $m$ instances of OD demand matrix ($X_1, X_2, X_2 \ldots \ldots \ldots , X_m$) to $m$ instances of flow ($F_1, F_2, F_3 \ldots \ldots \ldots F_m$), defined as follows,

$$\mathcal{F}([X_1, X_2, X_2 \ldots \ldots \ldots , X_m]; G(v, E, A_w)) = [F_1, F_2, F_3 \ldots \ldots \ldots F_m]$$

(2.2)

where, $A_w$ indicates the weighted adjacency matrix, $E$ indicates the set of links of the network and the vector $F_m$ contains the link flows for each link of the network for a given OD demand ($X_m$). In this formulation, OD demands, and network properties are input variables, while link flows are the target variables.
Figure 2.1 Data-Driven traffic assignment problem
<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{G}$</td>
<td>Transportation network</td>
</tr>
<tr>
<td>$v$</td>
<td>Set of nodes in $\mathcal{G}$ with size of $</td>
</tr>
<tr>
<td>$\mathcal{E}$</td>
<td>Set of links in $\mathcal{G}$ with size of $</td>
</tr>
<tr>
<td>$A_w \in R^{N \times N}$</td>
<td>Weighted adjacency matrix of $\mathcal{G}$, defined by Equation (2.1)</td>
</tr>
<tr>
<td>$I \in R^{N \times N}$</td>
<td>Identity matrix</td>
</tr>
<tr>
<td>$\bar{A}_w \in R^{N \times N}$</td>
<td>Neighborhood matrix defined by Equation (2.7)</td>
</tr>
<tr>
<td>$D_w \in R^{N \times N}$</td>
<td>Degree matrix of $\mathcal{G}$, a diagonal matrix where diagonal elements $(i, i)$ indicate the number of links coming out from a node</td>
</tr>
<tr>
<td>$L_w \in R^{N \times N}$</td>
<td>Laplacian matrix represents the structural properties of a network, defined by the equation (2.12)</td>
</tr>
<tr>
<td>$t_{i,j}^0$</td>
<td>Free flow travel time between nodes $i$ and $j$</td>
</tr>
<tr>
<td>$X \in R^{N \times N}$</td>
<td>OD demand matrix</td>
</tr>
<tr>
<td>$F \in R^E$</td>
<td>Flow vector contains flows for each link of the network for a given OD demand $X$</td>
</tr>
<tr>
<td>$\mathcal{P} \in R^{N \times N}$</td>
<td>Routing matrix indicates the probability of diffusion of flow from node $i$ to node $j$</td>
</tr>
<tr>
<td>$g_\theta$</td>
<td>Convolutional filter to learn the network features along with network function</td>
</tr>
<tr>
<td>$f(\cdot)$</td>
<td>Activation function</td>
</tr>
<tr>
<td>$\Theta \in R^{N \times N}$</td>
<td>Learnable parameters for the convolution filter</td>
</tr>
<tr>
<td>$W_q, W_F$</td>
<td>Learnable parameters for link flow estimation</td>
</tr>
<tr>
<td>$q$</td>
<td>Flow distribution matrix indicating the propagation of flow from a given node to different links of the network</td>
</tr>
<tr>
<td>$H$</td>
<td>Indicates the outputs from different layers of the proposed neural network architecture</td>
</tr>
</tbody>
</table>

All the bold letters denote a matrix
2.2.2 Graph Convolution Neural Network for Flow Pattern Learning

In our study, we develop a Graph Convolution based deep Neural Network (GCNN) architecture to assign traffic in the transportation network. GCNN generalizes the traditional convolutional neural network approach which uses a random filter (e.g., gaussian) to extract spatial correlations from the available features of a network. The network features are defined as a matrix, where each element includes specific information about the network (e.g., origin to destination demand). The convolutional operation multiplies a convolution filter with the network feature matrix to capture the cross-correlation among these spatial features. However, the problem with a convolutional neural network is that it considers the transportation network as an image, hence it does not capture how traffic states (e.g., flow) changes inside the network.

To determine the solution of a data-driven traffic assignment problem, we adopt the concept of graph convolution. We use graph convolution operation to learn the network properties and the flow diffusion process from origin nodes toward destination nodes. To estimate link flows we model how this flow diffusion process is contributing to link flows. In other words, the model considers how flows are coming to a specific link from adjacent nodes while diffusing from origin nodes towards destination nodes. In the proposed GCNN model a convolution filter is derived based on the network structure (position of the nodes and links) and the flow diffusion process inside the network. Hence, GCNN model captures changes in traffic states of a transportation network by modeling the flow diffusion from origin nodes toward destination nodes. As such, the GCNN model simultaneously learn the features (each node and link are embedded with valuable information) and function of the network.

Once the model learns the flow diffusion process, we can feed this information into a feed forward neural network to estimate the traffic flows at different links. In the following section we
describe the final architecture of the proposed deep neural network model to estimate traffic flows at different links.

**Figure 2.2 GCNN architecture for traffic flow pattern learning**

Figure 2.2 shows all the layers and matrix operations in each layer for the proposed GCNN architecture. In the first layer, we define the graph convolutional operation to learn the network properties and demand distribution for different nodes. We derive the graph convolution filter from weighted adjacency matrix where weights are assigned based on the free flow travel time of a link (see equation 2.1). We perform the convolution operation between the OD demand \((X)\) and graph convolution filter \((g_\theta)\). The convolutional filter \((g_\theta)\) represents the diffusion process of the traffic flows from the origin node towards destination nodes. While training the model, we will estimate the parameters \((\theta)\) for this convolutional filter. We define graph convolution layer as follows,
\[ H^1 = f_1(g_\theta \ast X) = f_1(\Theta g_\theta X) \]  

(2.3)

Here, \( f_1(.) \) denotes the nonlinear activation function for the convolution layer and \( H^1 \in \mathbb{R}^{N \times N} \) indicates the output from the graph convolution layer (1\textsuperscript{st} layer). From this layer, we obtain a convoluted demand matrix \((H^1)\) representing the flow diffusion process (from origin nodes towards destination nodes) inside a network.

The convoluted demand matrix is then fed into the 2\textsuperscript{nd} layer of the GCNN, where we model traffic flow distribution from origin nodes towards adjacent links. In this layer, we create a simple neural network model with parameters \( W_q \), which maps the convoluted demand matrix to a \( N \times E \) dimensional space (same size as the link-node adjacency matrix) via matrix multiplication. In this way, the GCNN captures how flow diffusion process will assign traffic at different links of the network. We define the 2\textsuperscript{nd} layer of the model as follows,

\[ H^2 = q = f_2(W_q H^1) \]  

(2.4)

Here, \( f_2(.) \) denotes the nonlinear activation function for the second layer and \( H^2 (= q) \in \mathbb{R}^{N \times E} \) indicates the output from the second layer representing distributed link flows from adjacent origin nodes \((N)\) of the network. Which means, each row of the matrix \( q \), indicates distributed link flows for all the links \((E)\) associated with an origin node.

Finally, the distributed link flow matrix \((q)\) is transposed and fed into the output layer (see Figure 2.2). Inside the transposed matrix \((q^T \in \mathbb{R}^{E \times N})\) each row indicates the distributed link flows for a given link from all the origin nodes \((N)\). In the output layer, we assign a linear activation function \((f_3(.)\)) with \( N \) parameters, which aggregates the distributed link flows and outputs assigned traffic flow for a given link. We define the output layer as follows,

\[ H^3 = F = f_3(W_F (H^2)^T) = f_3(W_F q^T) \]  

(2.5)
here, $f_3$ denotes the linear activation function and $H^3(= F) \in \mathbb{R}^E$ denotes the assigned traffic flows for all the links. From the output layer ($H^3$), we obtain link flows ($F$) for a given OD demand ($X$). So, the mathematical formulation of the GCNN model to estimate the link flows can be generalized as follows,

$$F = f_3((f_2(f_1(\Theta g_0 X)W_q)^T W_F))$$

(2.6)

here, we select the hyperbolic tangent function $tanh = \frac{e(x) - e(-x)}{e(x) + e(-x)}$ as the nonlinear activation function ($f_1(.) = f_2(.) = tanh$) for the model. In the following section, we describe the graph convolution operations and the parameters associated with graph convolution filter in details.

2.2.3 Modeling Traffic Flow using Diffusion Graph Convolution

In a transportation network, the traffic flow pattern changes in response to the changes in travel demand. We represent this relationship via flow diffusion process from origin nodes towards destination nodes. To capture the stochastic nature of traffic flow variation at a network level, we consider the flow diffusion process by a random walk (random movement into adjacent neighboring nodes) on network graph, $\mathcal{G}$ with restart probability $\alpha \in [0,1]$ and a state transition matrix $\tilde{D}_w^{-1}\tilde{A}_w$, where $\tilde{A}_w$ is the neighborhood matrix. In the neighborhood matrix, we add an identity matrix ($I$) with the adjacency matrix ($A_w$). By adding the identity matrix, we create a self-loop for each node; for a given diffusion step it will capture the traffic flows having same node as origin and destination, in other words it captures the traffic flow remains in the origin node rather than diffusing from origin node to destination nodes. The neighborhood matrix $\tilde{A}_w$ is defined as,

$$\tilde{A}_w = A_w + I$$

(2.7)

The restart probability indicates the probability of starting of a random walk from node $i$. From the starting node such random walks take multiple steps (diffusion steps, $K$) to traverse the
adjacent nodes until reaching the destination node \( j \). After many time steps, such diffusion process converges to a stationary distribution \( \mathbf{p} \in \mathbb{R}^{N \times N} \), where \( i \)th row of \( \mathbf{p} \) indicates the probability of flow diffusion from node \( i \) towards \( j \). The stationary distribution of the diffusion process can be represented as a weighted combination of infinite random walks on the graph [31] and be calculated in closed form,

\[
\mathbf{P} = \sum_{k=0}^{\infty} \alpha (1 - \alpha)^k (\overline{D}_w^{-1} \overline{A}_w)^k
\]  

(2.8)

where \( K \) is the diffusion step. In practice, we use a finite \( K \)-step truncation of the diffusion process and assign a trainable weight to each step [32, 33], based on that we can define the diffusion convolution over the network graph signal (e.g., OD demand) \( X \) as follows,

\[
g_\theta * X = \sum_{k=0}^{K-1} \Theta_k (\overline{D}_w^{-1} \overline{A}_w)^k X
\]  

(2.9)

If we consider 2 step diffusion process the above equation becomes,

\[
g_\theta * X = \Theta_0 X + \Theta_1 (\overline{D}_w^{-1} \overline{A}_w)^1 X
\]  

(2.10)

where, \( \Theta_0 \) and \( \Theta_1 \) are the parameters for each of the steps. While modeling the traffic flow pattern of transportation network, we can consider different number of diffusion steps to reach to a stationary distribution. However, as the network grows, this process becomes computationally expensive. Since for larger network the value for the diffusion step will be higher. Hence, such diffusion processes are only applicable for small scale networks, where flow diffusion occurs among the nearest neighbors [28, 34].

In our problem, we proposed an alternative approach to represent the diffusion process, rather than selecting diffusion step \( (K) \) values and assigning parameter to each step, we assign parameters \( (\Theta) \) to locally learn the stationary probability distributions (probability matrix). So,
while the training the model we estimate the parameters (Θ) to learn the stationary probability distribution for the flow diffusion process. In other words, we obtain a routing matrix [35] which indicates the probability of diffusion of flow from node \(i\) to node \(j\). The resulted diffusion convolution over the OD demand \(X\) can be written as follows:

\[
g_\theta \ast X = \Theta(D_w^{-1}A_w)X
\]  

(2.11)

where, \(\Theta \in R^{N \times N}\) are the parameters of the convolution filter and \(D_w^{-1}A_w\) represents the transition probability matrix of the diffusion process. In the demand matrix \(X\), each row indicates travel demand from origin node \(i\) to destination node \(j\). So, when we perform the matrix multiplication between the operator \(\Theta(D_w^{-1}A_w)\) and \(X\), we obtain a convoluted feature matrix which captures the influence all OD pairs on link flows associated with origin node.

We can also model the diffusion process using a normalized Laplacian matrix. Laplacian matrix better represents the structural properties of a network: the diagonal elements indicates the number of links originating at a given node, while the other elements indicate the connection between the origin and destination nodes. We define the Laplacian matrix as follows,

\[
\bar{L}_w = \bar{D}_w - \bar{A}_w
\]  

(2.12)

Normalizing the Laplacian matrix with degree matrix,

\[
\bar{D}_w^{-1}\bar{L}_w = \bar{D}_w^{-1}(\bar{D}_w - \bar{A}_w) = I - \bar{D}_w^{-1}\bar{A}_w
\]  

(2.13)

Now, the convolution over OD Demand matrix, \(X\) can be written as follows,

\[
g_\theta \ast X = \Theta(I - \bar{D}_w^{-1}\bar{A}_w)X
\]  

(2.14)
where, $\Theta \in R^{N \times N}$ are the parameters of the convolution filter, in other words, the coefficient matrix of the diffusion equation. During the training of deep learning model, we learn these parameters, which capture the flow diffusion process inside the network.

In this study we focus on a probabilistic approach to model the flow diffusion by estimating transition probability matrix in two different ways: using random walk on adjacency matrix (equation 2.11) and Laplacian matrix (equation 2.14). We also compare these approaches with spectral graph convolutional neural network [26] to learn traffic flow patterns. In spectral graph convolutional approach, the convolutional filter is estimated by decomposing the adjacency matrix into its eigenvalues to represent different properties of the graph such as strength of a node, shortest path distance etc.

2.3 Data Generation

Although we propose this method for real-world traffic data, recent sensing technologies are not densely distributed yet to provide us data necessary to test this approach. Especially, for large networks the OD demand variations are not available to us, though such OD demand data are possible due to the availability of mobile phone data[36, 37]. To test and verify our approach, we generate synthetic traffic data based on user equilibrium solutions of static traffic assignments over two networks: Sioux Falls network (24 nodes and 76 links) and East Massachusetts Network (74 nodes and 258 links) (Figure 2.3). We obtained the OD demand and information on network characteristics from [38]. To generate multiple OD demand, we multiplied the OD demand matrix by random factors collected from a uniform distribution which varies 0.1 to 1.0. To test our approach in different scenarios, we consider three conditions: uncongested, moderately congested, and fully congested. We generate 5000 OD matrices for each condition and solve both networks using the Frank Wolfe algorithm to obtain user equilibrium traffic flows. To represent the
prevailing traffic condition, we estimate the flow over capacity ratio. We assume that, for uncongested condition the flow-capacity ratio remains less than 0.5, for moderate condition the flow-capacity ratio varies between 0.4 and 0.8, and for uncongested condition, most of the cases the flow-capacity ratio is greater than 1.0. Figure 2.4 shows the traffic flow variations for different links of Sioux Falls Network.

Figure 2.3 Transportation networks (a) Sioux Falls network [38] (b) East Massachusetts network (US Census Bureau, 2015) [39]
(a) Uncongested Condition

(b) Moderately Congested Condition
2.4 Results and Discussion

We implemented all the models using PyTorch [40] library and train our model with dual NVIDIA Tesla V100 16GB PCIe GPU. Among the OD demand matrices, we use 70% \((n=3,500)\) for training, 20% \((n=1,000)\) for validation, and rest 10% \((n=500)\) of the data for testing the model. We train the model on training data and check the accuracy for the model on validation data set. Based on the validation accuracy we tune the hyperparameters such as learning rate, types of activation functions (i.e. tanh, sigmoid etc), maximum number of iterations. We also check whether the model is overfitting or not. Once the final model is fixed, we test it on the test data set. We calculate Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) as performance measures to check the accuracy of the implemented model. Performance metrics are defined as,
\[ RMSE = \sqrt{\frac{1}{m} \sum_{m=1}^{m} \sum_{E=1}^{E} \frac{(F_{m}^{E} - \hat{F}_{m}^{E})^2}{E}} \]  
\[ MAE = \frac{1}{m} \sum_{m=1}^{m} \frac{\sum_{E=1}^{E} |F_{m}^{E} - \hat{F}_{m}^{E}|}{E} \]  

In Table 2.2, we report the performance of model on test dataset. From the result we find that both diffusion convolution and spectral convolution have similar accuracy, which is expected, since spectral convolution operation is a special case of diffusion convolution [32]. Based on performance metrics values, we can conclude that the proposed approaches are performing reasonably well to capture the flow diffusion. RMSE and MAE values provides aggregated information (average over all the outputs) on the performance of the models, hence, we also estimated \( R^2 \) score. As shown in table 2.2, for each model the \( R^2 \) score is nearly 1, indicating the accuracy of the model to learn traffic flows inside the network. We have also plotted the relation between actual and estimated link flows for a given OD demand. Figure 2.5 and Figure 2.6 show that for both Sioux Falls and East Massachusetts networks the difference between actual and estimated link flow is quite low.
## Table 2.2 Model performances based on test dataset

<table>
<thead>
<tr>
<th>Network</th>
<th>Flow Propagation function</th>
<th>Minimum Flow</th>
<th>Maximum Flow</th>
<th>Mean Flow</th>
<th>MAE</th>
<th>RMSE</th>
<th>% Error Over Mean Flow</th>
<th>R² Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sioux Falls, Uncongested Free Flow Condition</td>
<td>Random Walk</td>
<td>642.8</td>
<td>7001.4</td>
<td>2447.4</td>
<td>8.6</td>
<td>11.9</td>
<td>0.350</td>
<td>0.9999</td>
</tr>
<tr>
<td></td>
<td>Laplacian Graph</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>spectral Graph</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>East Massachusetts</td>
<td>Random Walk</td>
<td>0</td>
<td>8220.1</td>
<td>1762.0</td>
<td>25.5</td>
<td>44.6</td>
<td>1.444</td>
<td>0.9991</td>
</tr>
<tr>
<td></td>
<td>Laplacian Graph</td>
<td></td>
<td></td>
<td></td>
<td>25.2</td>
<td>44.5</td>
<td>1.431</td>
<td>0.9991</td>
</tr>
<tr>
<td></td>
<td>spectral Graph</td>
<td></td>
<td></td>
<td></td>
<td>25.5</td>
<td>44.6</td>
<td>1.449</td>
<td>0.9991</td>
</tr>
<tr>
<td>Sioux Falls, Moderately Congested Condition</td>
<td>Random Walk</td>
<td>2562.9</td>
<td>16394.4</td>
<td>6704.5</td>
<td>23.2</td>
<td>31.7</td>
<td>0.345</td>
<td>0.9998</td>
</tr>
<tr>
<td></td>
<td>Laplacian Graph</td>
<td></td>
<td></td>
<td></td>
<td>23.1</td>
<td>31.7</td>
<td>0.345</td>
<td>0.9998</td>
</tr>
<tr>
<td></td>
<td>spectral Graph</td>
<td></td>
<td></td>
<td></td>
<td>23.4</td>
<td>32.1</td>
<td>0.350</td>
<td>0.9998</td>
</tr>
<tr>
<td>East Massachusetts</td>
<td>Random Walk</td>
<td>0</td>
<td>10858.7</td>
<td>2337.9</td>
<td>33.6</td>
<td>59.5</td>
<td>1.436</td>
<td>0.9991</td>
</tr>
<tr>
<td></td>
<td>Laplacian Graph</td>
<td></td>
<td></td>
<td></td>
<td>33.5</td>
<td>59.4</td>
<td>1.431</td>
<td>0.9991</td>
</tr>
<tr>
<td></td>
<td>spectral Graph</td>
<td></td>
<td></td>
<td></td>
<td>33.6</td>
<td>59.5</td>
<td>1.438</td>
<td>0.9991</td>
</tr>
<tr>
<td>Sioux Falls, Congested Condition</td>
<td>Random Walk</td>
<td>4489.6</td>
<td>23437.3</td>
<td>10408.7</td>
<td>36.0</td>
<td>48.4</td>
<td>0.346</td>
<td>0.9998</td>
</tr>
<tr>
<td></td>
<td>Laplacian Graph</td>
<td></td>
<td></td>
<td></td>
<td>35.3</td>
<td>47.4</td>
<td>0.338</td>
<td>0.9998</td>
</tr>
<tr>
<td></td>
<td>spectral Graph</td>
<td></td>
<td></td>
<td></td>
<td>36.9</td>
<td>49.2</td>
<td>0.355</td>
<td>0.9999</td>
</tr>
<tr>
<td>East Massachusetts</td>
<td>Random Walk</td>
<td>0</td>
<td>12174.8</td>
<td>2698.7</td>
<td>38.8</td>
<td>68.9</td>
<td>1.438</td>
<td>0.9991</td>
</tr>
<tr>
<td></td>
<td>Laplacian Graph</td>
<td></td>
<td></td>
<td></td>
<td>38.7</td>
<td>68.8</td>
<td>1.439</td>
<td>0.9991</td>
</tr>
<tr>
<td></td>
<td>Spectral Graph</td>
<td></td>
<td></td>
<td></td>
<td>39.3</td>
<td>69.1</td>
<td>1.4575</td>
<td>0.9991</td>
</tr>
</tbody>
</table>
From the results, we get two evidence: first, a neural network can capture the traffic assignment of a network without any prior knowledge on user behavior; second, we can achieve better accuracy with an appropriate representation of the physical process of flow propagation.

(a) Uncongested Condition                       (b) Moderately Congested Condition

(c) Congested Condition

Figure 2.5 Comparison between actual and estimated link flows for a given OD demand

(Sioux Falls Network)
2.4.1 Result interpretation

To understand how well the model has learned the flow propagation for the given networks, we perform network topology analysis for the Sioux Falls network based on the betweenness centrality of the nodes. Betweenness centrality of a node indicates the fraction of the total number of shortest paths passing through that node [41], which means a node with a higher value of betweenness centrality will have a higher number of shortest paths incidence on that particular node.
For a given OD demand matrix, to estimate betweenness centrality values we need to find the shortest paths for all pairs of origin-destination nodes at user equilibrium. However, the shortest path for a pair of origin and destination nodes depends on link travel times. In our analytical solutions (i.e., using Frank-Wolfe algorithm), we use BPR travel time function (see equation 2.17) to update link travel times. Based on the updated travel time, we find the shortest paths for assigning traffic to the network. These steps continue iteratively prior to reaching an user equilibrium solution, when travel time for all the used paths remain same for a given O-D pair. Hence, we use the same function to estimate user equilibrium travel time \( t_{i,j} \) for all the links of the network.

\[
t_{i,j} = t_{i,j}^0 \left( 1 + 0.15 \left( \frac{f_{i,j}}{c_{i,j}} \right)^4 \right)
\]  

(2.17)

where \( \frac{f_{i,j}}{c_{i,j}} \) indicates the flow-capacity ratio for a given link. Based on the estimated links’ travel time from equilibrium link flows we find the shortest paths and estimate the betweenness centrality for all the nodes. We apply this approach on all the training OD demand samples. Figure 2.7 shows the distribution of betweenness centrality values for each node of the Sioux Falls Network across different traffic conditions. From the Figures we find that except nodes 1, 2, 9, and 23, all the nodes have higher betweenness centrality values. Nodes 6, 8, 12, 15, 16, and 18 are the most critical ones for the Sioux Falls network. Moreover, we find that for congested condition (i.e. higher demand), variations of betweenness centrality values are higher compared to moderately and uncongested conditions.
(a) Congested Condition                          (b) Moderately congested Condition

Figure 2.7 Variations of Betweenness Centrality for different nodes at different traffic condition (Sioux Falls Network)

From the BPR function, we find that, if flow capacity ratio is less than 1.0, there will be no significant change in travel time with link flow variations or demand variations. However, when the flow capacity ratio is greater than 1.0, travel time will vary significantly with demand variations (i.e., link flow variations). Consequently, the shortest paths will change abruptly (i.e., shortest path are not stable) leading to significant variations in betweenness centrality for all the nodes. In our case, for congested condition, the flow capacity ratio mostly varies from 1.0 to 2.0, thereby we observe a significant variation of betweenness centrality for different nodes. Whereas for
moderately congested and uncongested conditions, since the flow capacity ratio is mostly less than 1.0 so we, do not see significant variations in Betweenness Centrality.

In the proposed model, we assign parameters \( W_q \) to learn the flow propagation from nodes into adjacent neighboring links. We assume that the weight parameters associated with critical nodes will be higher and will vary significantly due to the changes in betweenness centrality of nodes. In other words, the weight parameter associated with a node is likely be correlated with the betweenness centrality value of the node.

In Figure. 2.8, we plot the weight distributions for node-link flow propagation at different traffic conditions. Since for uncongested and moderately congested conditions the variations of betweenness centrality for different nodes is similar, the distribution of weight parameters \( W_q \) are also similar. In both cases, the critical nodes 6, 8, 15, and 16 have high positive weights (Figure. 2.8 (a) & (b)). For congested condition, the variation of betweenness centrality over different nodes are higher, thereby the weight parameters vary significantly compared to uncongested and moderately conditions (Figure. 2.8(c)). In a congested condition, the model cannot identify the critical nodes from all nodes to pass the traffic efficiently. This could be a possible reason that the model does not give higher positive weights for critical nodes similar to uncongested and moderately congested conditions.
Figure 2.8 Distribution of weight ($w_q$) over different traffic conditions

(a) Uncongested Condition

(b) Moderately Congested Condition

(c) Congested Condition
2.4.2 Stability of the Solution

We train the model using mean square error as the loss function. At each iteration, the model estimates the mean square error for the estimated flows ($\hat{F}_m^E$) and the actual flows ($F_m^E$) of the network. Afterward, the gradient of the loss function is backpropagated to adjust the weights to reduce loss function value. The loss function can be defined as,

$$L_m = Loss(F_m^E, \hat{F}_m^E)$$

(2.18)

$$MSE = \frac{1}{m} \sum_{m=1}^{m} \sum_{E=1}^{E} (F_m^E - \hat{F}_m^E)^2$$

(2.19)

where, $Loss(.)$ is the function to estimate the error between the actual ($F_m^E$) and estimated values ($\hat{F}_m^E$) and $E$ denotes the set of links for the network. In this study, we estimate mean square error (MSE) as a loss function.

To check the stability of solution, we observed the training and test loss values for the model (Figure 2.9). We train each model for 10,000 iterations to check variation of train and validation loss values. We find that it takes about 2000 iterations for the model to converge to a stable solution, after that there are merely any variations in loss values. Moreover, after 10,000 iterations the loss function value for the validation data gradually start increasing. From 2,000 to 10,000 iterations the difference in MSE values varies from 4 to 9, indicating a stable solution with minimal variance. We experimented with two different optimizers to train the model: root mean square propagation (RMSProp) and adaptive moment estimation (ADAM) optimizer. Among these two, RMSProp takes less iterations (~1500 iterations) to converge (i.e. similar train and validation error) compared to ADAM optimizer (~2000 iterations). However, ADAM optimizer gives more stable solutions, which means MSE values for train and validation data almost remain
same after convergence (i.e., after 2000 iterations). Whereas, for RMSProp optimizer we observe slight variations in MSE values for both train and validation data samples even after convergence (i.e., after 1500 iterations).

We also check the computation time required to train the models. It takes about 19 minutes to train the models on Sioux Falls Network for 10,000 iterations, while for East Massachusetts Network it takes 30 minutes. So, our approach performs reasonably well to estimate network level traffic flows with less computation time.

(a) Congested Condition
Figure 2.9 Loss function values for Laplacian spectral Graph
2.5 Conclusions

This chapter presents a data-driven formulation of traffic assignment problem based on learning traffic flow patterns of a transportation network from origin destination (OD) travel demand variations. Adapting graph convolution approach, we develop a deep learning architecture to solve this traffic assignment problem by capturing the diffusion of OD demand inside the network. To efficiently represents the diffusion process of multiple OD demands from nodes to neighboring links, we customize the traditional graph convolutional neural network and introduce the concept of learning-based assignment (i.e. routing matrix) of OD demands to generate link flows. Finally, we provide experimental evidence on the validity of the approach by training the model to learn the user equilibrium traffic flows for Sioux Falls and East Massachusetts networks. The experiment results show that the implemented GCNN model can capture the user equilibrium traffic flow of the network very well with less than 2% mean absolute difference between the actual and estimated link flows under varying congested conditions. Moreover, when the training of the model is complete, it can instantaneously determine the traffic flows of a large-scale network. Hence this approach can overcome the challenges of deploying traffic assignment models over large-scale networks. Furthermore, this method is completely data-driven without requiring any assumption on user behavior. Thus, it will improve the reliability and stability of traffic assignment solutions.
CHAPTER 3: A DATA-DRIVEN NETWORK MODEL FOR TRAFFIC VOLUME PREDICTION AT SIGNALIZED INTERSECTIONS

3.1 Introduction

Network-wide high-fidelity traffic prediction can benefit transportation systems management and operations including measuring traffic signal performance, optimizing signal timing plans, and managing incidents. Typically, traditional mathematical or simulation-based network modeling approaches have been applied to estimate traffic flows [4, 5]. Although these approaches produce reasonable solutions for traffic prediction problems, the complexity and computation time required to implement such model make them less suitable for real world application such as real-time link flow or intersection-level traffic volume prediction. In addition, these frameworks need to be modified to ingest large-scale data (such as automated traffic signal performance measures) available from intersections.

Although advancement of traffic sensing technologies has created an opportunity to overcome these challenges and implement deployable modeling approaches to predict traffic at higher resolution such as intersection level (i.e., direction wise and movement wise). However, large-scale network level traffic forecasting is more challenging due to higher computational complexity because of network size. For this purpose, a robust prediction model is required with the ability to capture spatial correlation of traffic among adjacent roads and learn traffic movement patterns from high resolution data.

In this chapter, we present a new modeling framework for intersection level movement volume prediction for a large-scale network. We develop a graph convolution based deep learning model to predict traffic movement volume by capturing the correlation between demand features and traffic flow patterns of a transportation network. To test the model, we have collected
Automated Signal Performance Measure (ATSPM) data and zonal-level travel demand data for Seminole County, Florida. This particular research has made several contributions:

- It develops a data pipeline incorporating extensive data assessment approaches to extract and process traffic movement volumes from ATSPM data.
- It develops a new method for intersection-level traffic prediction considering the correlation between travel demand and flow inside a transportation network; and
- It provides empirical evidence on the performance of a deep learning-based model for traffic prediction using real-world large-scale traffic signal performance data.

3.2 Literature Review

Understanding traffic evolution and congestion propagation for an entire road network rather than a single road will be more helpful for traffic managers in proactive decision making [42]. However, large-scale network level traffic forecasting is more challenging due to higher computational complexity incurred by the network size and topology; which requires a robust prediction model, with the ability (i) to capture the spatial correlation of traffic in interconnected roads, and (ii) to predict traffic for a long-term to reflect congestion propagation. But traditional traffic prediction models [43, 44] consider traffic state variables as sequential data, thus cannot deal with high dimensionality of the data to learn spatial correlation.

Convolutional LSTM methods are the initial attempt to model the spatial and temporal correlation among the traffic states for network level traffic prediction. A few studies [45] have implemented the convolutional LSTM model for network level traffic speed prediction. Although this model outperforms existing state of art data-driven model, it does not consider stochastic traffic flow dynamics (i.e. flow propagation) while extracting spatial correlation among network traffic. Recently, graph convolution neural network [26, 46, 47] has been emerging as a new
approach to overcome the limitation of convolutional neural networks in traffic prediction problem. Graph convolutional neural network approaches utilize the concepts of graph theory along with deep neural network architectures to model the stochastic traffic dynamics inside a network. These approaches aim at learning the interactions between roadways in the traffic network to forecast network-wide traffic states. However, the application of such a neural network architecture hardly exists for a large-scale transportation network. A few studies has utilized the concept of graph convolution to represent traffic network as a generalized graph for traffic state prediction [27, 28]. These studies, however, have focused on learning network-wide short-term correlations among traffic states (e.g., speed) to predict future states 5 to 20 mins ahead of time.

Data-driven methods for intersection level traffic prediction mostly involve traffic flow [48], movement volume prediction [49–51] and traffic signal queue length prediction [44, 52, 53]. These studies are highly data intensive; previous studies either used a simulation-based or a hybrid approach to develop intersection-level traffic prediction models. For instance, Chang and Su [52] developed a data-driven neural network model for predicting queue length at short time step (3s). They used the data from simulation experiments to train the model for queue prediction. Lee et al. [53] developed a deep learning model for queue length estimation. They relied on traffic simulations to generate the training data and used real-world driving data from the Federal Highway Administration’s Next Generation Simulation (NGSIM) program to test the approach. However, one limitation is that these approaches are based on isolated intersections without considering the coordination among multiple intersections.

Recently, Alajali et al. [48] applied gradient boosted decision tree based model to predict intersection traffic volume for large scale network which covers intersections at central business district (CBD) area of Melbourne, Australia. This study proposed an online and offline training
approach to deal with the limitation in computation power for large scale data. However, the proposed method is limited to aggregate traffic volume prediction at intersection level rather than traffic movement volumes. A recent study by Li et al. [49] proposed a deep learning method to predict intersection level traffic movement volume. This study utilizes Convolutional LSTM model to capture spatiotemporal dependency among network-wide traffic states considering traffic network as an image. Thereby fails to capture the stochastic traffic flow dynamics of the network. Moreover, the proposed approach does not consider travel demand features, thus limited to only short-term traffic movement prediction (i.e., 5-15 min ahead of current time).

In summary, we find that most of previous studies adopted deep learning models without considering directional traffic volume. Moreover, capturing the spatiotemporal dependency of traffic for interconnected roadway segments can improve the model performance compared to traditional times series-based approaches. However, still there exists several research gaps in application of deep learning methods for intersection-level traffic movement volume prediction. First, these deep learning methods have not been tested for network scale intersection level traffic movement volume prediction, rather applied over isolated intersections or at a corridor level. Second, although graph theoretic approach has been applied for detector level or segment wise traffic state prediction, no study has considered traffic dynamics and tested its influence on model accuracy. Third, previous studies focused on short-term or cycle-level traffic prediction, thus, do not consider any demand related features to account for long-term demand variations. However, for predicting traffic over a long-term period (>1 hr.) we need to add the information on travel demand variation over different periods (peak hour, off peak hours etc.).
3.3 Problem Formulation

We present a data-driven approach of learning intersection-level traffic movement patterns of a transportation network given that information on travel demand and corresponding traffic movement volumes are available. Instead of estimating traffic movement patterns using traditional traffic assignment models, here we implement the idea of learning those movement patterns from large-scale training data. Adopting the concept of graph convolution, we develop a deep learning model to capture the cross correlation among spatiotemporal traffic features to predict traffic movement over a long-term sequence.

To implement the model, we represent the transportation network as a graph where each node indicates an intersection, and the edges indicate the shortest path distance between two intersections. Let, $\mathcal{G}(v, E, A)$ is an undirected graph, where $v$ denotes the set of nodes (i.e., intersection) and $E$ denotes the set of links between nodes $(i, j)$. $A$ represents the connectivity between nodes as a weighted adjacency matrix, where weights are based on distance between any two nodes $(i, j)$, defined as follows:

$$A(i, j) = \begin{cases} d_{i,j} & \text{if } i \neq j \\ 0, & \text{otherwise} \end{cases} \quad (3.1)$$

where, $d_{i,j}$ denotes the distance time between the origin $i$ and the destination $j$ nodes. The connectivity inside an adjacency matrix detects which neighboring nodes $(j)$ will be influenced by the traffic condition at a given node $(i)$. Moreover, in a timeseries problem the existing traffic condition at a given node $(i)$ will also influence its future traffic condition, which means each node is temporally self-influenced. This is represented by adding an identity ($I$) matrix with the adjacency matrix which ensures that nodes are self-accessible,

$$\bar{A} = A + I \quad (3.2)$$
The proposed data-driven traffic prediction model aims to learn intersection level traffic movement volume of a transportation network over multiple future time-steps based on capturing the spatiotemporal correlation among traffic features at different intersection. In other words, the framework captures how traffic at a given intersection influences the traffic condition at other intersection based on the distance between two intersections. Thus, we feed the model with information of two aspects: (i) a weighted adjacency matrix ($\tilde{A}$) indicating the connectivity and distance between intersections and (ii) node level features ($X_t$) representing traffic demand and traffic state variations.

The traffic prediction problem aims to learn a function $F(.)$ that maps $l$ instances of input sequence ($[X_{t-l}, X_{t-l+1} ..., X_t]$) to predict intersection-level traffic movement volume ($F_{t+1}, F_{t+2} .... F_{t+p}$) for the entire network. We define the problem as follows:

$$F([X_{t-l}, X_{t-l+1} ..., X_t]; G_t(v, \mathcal{E}, \tilde{A})) = [F_{t+1}, F_{t+2} ... F_{t+p}]$$  \hspace{1cm} (3.3)

where, $l (= 0,1,2 ... l)$ and $p (= 1,2,3 ... p)$ indicates the input and output sequence; $F_{t+p} \in \mathbb{R}^{N \times 12}$ indicate the traffic movement volumes for the entire network at time $t + p$. Each row of the matrix $F_{t+p}$ indicates all the possible movements (i.e., left, through, and right) at each approach (e.g., four approaches) of an intersection. Description for all the notations associated with the model development is included in Table 3.1.
Table 3.1 Description of the notations associated with the model development

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathcal{G} )</td>
<td>Transportation network</td>
</tr>
<tr>
<td>( v )</td>
<td>Set of nodes in ( \mathcal{G} ) with size of (</td>
</tr>
<tr>
<td>( \mathcal{E} )</td>
<td>Set of links in ( \mathcal{G} ) with size of (</td>
</tr>
<tr>
<td>( A \in \mathbb{R}^{N \times N} )</td>
<td>Weighted adjacency matrix of ( \mathcal{G} ), defined by Equation (1)</td>
</tr>
<tr>
<td>( I \in \mathbb{R}^{N \times N} )</td>
<td>Identity matrix</td>
</tr>
<tr>
<td>( \overline{A} \in \mathbb{R}^{N \times N} )</td>
<td>Neighborhood matrix defined by Equation (2)</td>
</tr>
<tr>
<td>( \mathbf{D} \in \mathbb{R}^{N \times N} )</td>
<td>Degree matrix of ( \mathcal{G} ), a diagonal matrix where diagonal elements ((i, i)) indicate the number of links connected with a node</td>
</tr>
<tr>
<td>( d_{ij} )</td>
<td>Distance between nodes ( i ) and ( j )</td>
</tr>
<tr>
<td>( l )</td>
<td>Input time sequence length ((0,1, \ldots, l))</td>
</tr>
<tr>
<td>( c )</td>
<td>Number of input features</td>
</tr>
<tr>
<td>( X_t \in \mathbb{R}^{N \times c} )</td>
<td>Contains all the traffic features associated with each node ((i)) of the network</td>
</tr>
<tr>
<td>( g )</td>
<td>Graph Convolutional filter to learn the congestion propagation inside the network</td>
</tr>
<tr>
<td>( f(\cdot) )</td>
<td>Activation function</td>
</tr>
<tr>
<td>( W_{gc} \in \mathbb{R}^{N \times N} )</td>
<td>Learnable parameters for the convolution filter</td>
</tr>
<tr>
<td>( H )</td>
<td>Indicates the outputs from different layers of the proposed neural network architecture</td>
</tr>
<tr>
<td>( p )</td>
<td>Prediction horizon ((1, \ldots, p))</td>
</tr>
<tr>
<td>( m )</td>
<td>Set of movement types (i.e. North left, etc.) with size (</td>
</tr>
<tr>
<td>( F_{t+p} \in \mathbb{R}^{N \times m} )</td>
<td>Intersection level traffic movement volumes for entire network</td>
</tr>
</tbody>
</table>

All the bold letters denote a matrix
3.4 Methodology

3.4.1 Graph Convolution for Spatial Dependency Modeling

In a transportation network, the nodes are interconnected so traffic condition at a given node will impact the traffic in the neighboring nodes. This is because of the propagation of traffic from the origin nodes towards the neighboring nodes until the destination nodes are reached. So, any node that is located within the shortest path distance of an origin node will be influenced by flow propagation process. Graph convolutional neural network can capture these intrinsic dynamics of the network (i.e. flow propagation), thus captures the spatial cross correlation among the nodes.
neighboring nodes based on their position inside the network (Figure 3.1(a)). Representation of the flow propagation process requires a graph convolution filter which is derived from the adjacency matrix. In a graph theoretic approach, graph adjacency matrix is decomposed into its eigenvalues to represent the structural properties of a graph such as strength of a node (e.g. node level features), shortest path between two nodes etc. Such representation while feed into deep learning model suffers from exploding or vanishing gradient problem due to sparsity in eigenvalues’ distribution. To overcome this exploding or vanishing gradient problem, Kipf and Welling [26] proposed a normalization technique to represent a graph and its intrinsic dynamics known as spectral graph. Based on spectral graph representation method the convolution operation to capture the spatial correlation can be defined as follows,

\[ g_{ct} = (W_{gc} \odot \overline{D}^{-\frac{1}{2}} \overline{AD}^{-\frac{1}{2}})X_t \]  

where, \( g_{ct} \) indicates the convoluted feature matrix and \( W_{gc} \) indicates the parameters for the convolution filter. \( \overline{A} \) and \( \overline{D} \) indicates the adjacency matrix and degree matrix respectively and the expression \( \overline{D}^{-\frac{1}{2}} \overline{AD}^{-\frac{1}{2}} \) indicates the symmetrically normalized adjacency matrix. \( X_t \) indicates node level features, representing traffic demand variations. The convoluted feature matrix represents the state transition of the network, in other words changes in traffic dynamics inside the network.

### 3.4.2 LSTM Model for Temporal Dependency Modeling

In traffic prediction problems, an LSTM [54–56] model is applied to capture the temporal correlation among traffic features. The basic difference between LSTM model and other neural network model is that in a simple neural network model, a hidden state is stored as single vector or matrix whereas in a LSTM model the hidden state consist of two vectors a short-term state \( h_t \) and a long-term state \( c_t \) (Figure 3.1 (b)). To capture the short term correlation (i.e. hourly pattern) between traffic features of two consecutive timesteps the short term state is recursively fed into
LSTM unit. Moreover, LSTM keep the long-term information (such as period or seasonal traffic variations) as it passes over multiple time steps. At each time step (t) the hidden states \((h_t, c_t)\) are continuously updated by four fully connected neural network layers known as input \(i(t)\), forget \(f(t)\), cell \(g(t)\) and output \(o(t)\) layers. In the final time step the hidden states \((h_t, c_t)\) are fed into the output layer to get the final output \((y(0))\).

### 3.5 Modeling Frameworks

We proposed two different graph convolution based neural network architectures to model the traffic movement volumes over multiple temporal sequences: Graph Convolutional LSTM (GCN-LSTM) and Graph Convolutional Encoder Decoder LSTM (GCN-Encoder-Decoder). In the following sections, the proposed modeling framework is described.

#### 3.5.1 Graph Convolutional LSTM

In the Graph Convolutional LSTM (GCN-LSTM) architecture, we stack the graph convolution and LSTM layer to capture both spatial and temporal dependency of traffic features. The model takes intersection level traffic features \((X_t)\) such as hourly volume, zonal level trip attraction and production, characteristics of the built environment as inputs and outputs the predicted traffic movement volumes. Figure 3.2 (a) shows different components of the model as it unrolls over multiple time steps (i.e. length of input sequences). At each time step, the model performs a graph convolution operation over the input data \((G_t)\) and feeds the output \((g_c_t)\) into the LSTM model. After taking the input from the graph convolution \((GC)\) layer, it updates the hidden states \((h_t, c_t)\).

In this architecture, the short-term state \((h_t)\) at the final time step \((t)\) is linearly transformed using a fully connected layer to get the predicted \((y_t)\) traffic movement volumes over multiples sequences.
(a)
Figure 3.2 Modeling framework (a) Graph Convolutional LSTM (b) Graph Convolutional Encoder Decoder LSTM

3.5.2 Graph Convolutional Encoder Decoder LSTM

In this architecture, the outputs from the graph convolutional layer are fed into an encoder-decoder LSTM architecture. Similar to previous architecture, the encoder LSTM captures the short-term and long-term dependencies of traffic features as it unrolls over multiple time step, thus it updates the hidden states \((h_t, c_t)\). However, to get the final outputs over multiple temporal sequences, we use a decoder LSTM (Figure 3.2 (b)). At the first-time step, the decoder LSTM takes the hidden
states \((h_t, c_t)\) and input state \((g c_t)\) from the final encoder LSTM step \((t)\) to predict traffic over 1-step ahead \((F_{t+1})\). In the following time steps, it recursively takes the predicted traffic volumes \((F_{t+1})\) and hidden states \((h_{t+1}, c_{t+1})\) from the previous step to predict the traffic volumes in the next step \((F_{t+2})\). The decoder LSTM iteratively performs this operation to generate the whole output sequence.

### 3.6 Data Collection and Preprocessing

#### 3.6.1 Traffic Movement Volumes

In Seminole County, majority of the signalized intersections are equipped with advanced traffic signal controllers on the arterials and each signal provides Automated Traffic Signal Performance Measures (ATSPM) (Figure 3.3), which includes traffic movement volumes, signal timing, traffic queue volume etc. We extracted high-resolution event-based signal data for Seminole county from January 1, 2016 to December 31, 2016; in total we have collected data for 253 intersections. From the raw dataset, we mainly extracted the traffic movement volumes for different direction and movement types.
The raw data collected from traffic detectors are subjected to errors. Several factors such as detector’s malfunctioning, false encoding during storing the data into the server, overlapping of multiple entries, duplicate entries, bad weather conditions etc. can cause errors. Therefore, before proceeding to any data analysis, we need an extensive data cleaning and quality checking. Figure 3.4 shows the framework for the data processing steps.

To check the quality of the data, we followed several steps starting with checking the percentage of missing data for different detectors. We consider the detectors having a higher percentage of missing values as unreliable ones. Moreover, data imputation is not feasible for the detectors with too many missing values as it will produce unrealistic data distribution. Considering this issue, we retain the detectors having missing values less than 20% of total data samples. We apply three techniques to check the quality of the data and detecting outliers.
First, we compare vehicle per hour for each movement types (i.e., left, through, right) with Federal Highway Administration (FHWA) guidelines on maximum capacities at signalized intersection. According to the FHWA guidelines for left and right turn movement the capacities vary from 150-350 veh/hr, while for through movement the capacity varies from 1600-2100 veh/hr depending on number of available lanes [57]. We observe the distribution of the hourly turning movement volume for different movement types. Almost all the data samples, except a few (less than 0.1%), have hourly volume less than capacity. We consider the samples with values greater than capacity as outliers.

**Figure 3.4 Data processing framework**
Second, we apply isolation forest algorithm [58] to detect the outlier from temporal pattern of the traffic movement volume. The algorithm learns traffic pattern for different hours of the day and days of the week and isolates the outliers which show unusual pattern. Third, we check if the turning movement volume remains within the range between \((Q1 - 1.5 \times IQR)\) and \((Q3 - 1.5 \times IQR)\), where \(Q1\) and \(Q3\) indicates the first and third quartile and IQR indicates the interquartile range \((Q3 - Q1)\).

Finally, we use a technique known as multivariate iterative imputation [59] adapting Bayesian ridge regression as estimator to impute the missing values and outliers. To fit the estimator, we use time of the day (hour), day of the week (day), and volume with missing values as inputs. For each imputation the algorithm takes a sample from gaussian posterior of the fitted estimator. We use Python scikit learn [60] library to implement the algorithm. The details about the data imputation algorithm are provided in reference [59]. Figure 3.5 shows the distribution of through movement volumes before and after data processing. After cleaning the data, we have in total 196 intersections in our final data set. Moreover, we only keep the data in between 6 am to 12am, most of cases from 12 am to 6 am the traffic activity within this region is either zero or nearly zero. We have also removed the data for special events and holidays such as hurricane evacuation period (September 28, 2016 – October 10, 2016), thanksgiving holiday etc.
3.6.2 Travel Demand Features

To generate the travel demand for the study area, we employ Central Florida Regional Planning Model (CFRPM 6.1) daily model which outputs average weekday trip production and trip attraction at the Traffic Analysis Zone (TAZ) level by trip purpose and special generators. Our study area (Seminole County, Florida) includes 230 TAZs, so, we can aggregate the weekday trips to estimate total production and total attraction for each of these 230 zones. However, each TAZ has multiple intersections within its boundary. Hence, to link this demand information with operational characteristics of an intersection, we need to partition these TAZs into a finer spatial resolution.

Considering this issue, we develop a technique to partition the TAZs into multiple subzones to generate demand at a finer resolution. Based on the frequency of the intersections within each TAZ, we divide them into 5 subzones. To create the subzones, we follow a three steps process: Firstly, we identify the centroid of each zone and the points in the perimeter that is used to create the polygon, i.e., TAZ boundary. Secondly, depending on the number of subzones (i.e. 5) for each TAZ, we identify the group of points that needed to be connected with the centroid. Finally, we

![Figure 3.5 Distribution of data samples based on vehicles per hour per lane](image)

3.6.2 Travel Demand Features

To generate the travel demand for the study area, we employ Central Florida Regional Planning Model (CFRPM 6.1) daily model which outputs average weekday trip production and trip attraction at the Traffic Analysis Zone (TAZ) level by trip purpose and special generators. Our study area (Seminole County, Florida) includes 230 TAZs, so, we can aggregate the weekday trips to estimate total production and total attraction for each of these 230 zones. However, each TAZ has multiple intersections within its boundary. Hence, to link this demand information with operational characteristics of an intersection, we need to partition these TAZs into a finer spatial resolution.

Considering this issue, we develop a technique to partition the TAZs into multiple subzones to generate demand at a finer resolution. Based on the frequency of the intersections within each TAZ, we divide them into 5 subzones. To create the subzones, we follow a three steps process: Firstly, we identify the centroid of each zone and the points in the perimeter that is used to create the polygon, i.e., TAZ boundary. Secondly, depending on the number of subzones (i.e. 5) for each TAZ, we identify the group of points that needed to be connected with the centroid. Finally, we
continue this process until reaching to the last point. In total we create 1150 subzones from 230 TAZs. Figure 3.6 show the TAZs and generated Subzones for the study area. From the CFRPM model we estimate average weekday trip attraction and production at a subzone. Afterwards, we expand the daily demand into hourly demand using hourly distribution factors provided by CFRPM (see [61] for more details).

![Figure 3.6 Traffic analysis zones (TAZs) and Sub zones for the analysis](image)

We have also extracted exogenous variables including built environment and land-use characteristics for each subzone. Built environment characteristics variables are processed from NAVSTREET data and include number of restaurants, shopping centers, business centers, entertainment establishments and educational institutions. Land-use characteristics are processed using high resolution parcel level land-use data sourced from Florida Department of Revenue. Each parcel is assigned a unique ID (Parcel ID) linking it with equivalent parcel level attribute information such as property/feature value, land value, land area in square feet, land-use codes (DOR-UC), owner name, owner address, physical address, physical zip code, building details and so on contained in the Name-Address-Legal (NAL) file. In this study we consider five land use categories: Residential, Retail/Office, Industrial/Manufacturing, Institutional/Infrastructure, Recreational etc.
3.7 Experiments

3.7.1 Input features and Graph Representation

In this section we discuss about spatiotemporal features extraction technique from the data. We also discuss about the graph representation from geolocation of the intersections.

**Input features in the data samples** \((X_t)\): We have extracted different types of features to represent the travel demand and built environment characteristics corresponding to each intersection. Since the spatial variations in built environment characteristics are more subtle, we use natural logarithm of the areas (acre) corresponding to different built environment types (Table 3.2), which scales the subtle changes and make it more prominent. Finally, we merge these variables with hourly trip attraction and trip production. To feed the model with short term travel demand variations, we use aggregated intersection level hourly traffic volume. We also use temporal features such as time of the day and days of the week to capture the seasonality inside the data.

In total we have 11 features; we formulate the traffic data sample as \([\text{number of samples} (n), \text{input time sequence} (l), \text{number of nodes} (N), \text{input features} (c)]\). Since, we have collected the data from 196 detectors, so the number of nodes, \(N = 196\). We select 6-hour input data sequence to predict traffic for next 4 hour, so input time sequence length \(l = 6\) and prediction horizon length, \(p = 4\). In total we have \(c = 11\) input features: days of the week (i.e. Saturday, Sunday etc.), time of the day as hour (1 to 24 hr), intersection specific aggregated traffic volume, hourly trip attraction, hourly trip production, proportion of highway among total roadways of a zone. We have 5661 data samples from a one-year period (2016). Finally, the input data and target data samples have the shape as \([5661, 6, 196, 11]\) and \([5661, 4, 196, 12]\), respectively.
Table 3.2 Description of input features

<table>
<thead>
<tr>
<th>Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_d$</td>
<td>Days of the week (i.e. Saturday, Sunday etc.)</td>
</tr>
<tr>
<td>$t_{hr}$</td>
<td>Time of the day as hour (1 to 24 hr)</td>
</tr>
<tr>
<td>$q_t$</td>
<td>Intersection specific aggregated traffic volume at time $t$</td>
</tr>
<tr>
<td>$T_A$</td>
<td>Hourly Trip Attraction</td>
</tr>
<tr>
<td>$T_P$</td>
<td>Hourly Trip Production</td>
</tr>
<tr>
<td>$HW_{prop}$</td>
<td>Proportion of Highway among total roadways of a zone</td>
</tr>
<tr>
<td>$B_{ind}$</td>
<td>Ln(industrial)</td>
</tr>
<tr>
<td>$B_{rec}$</td>
<td>Ln(recreational)</td>
</tr>
<tr>
<td>$B_{ret}$</td>
<td>Ln(retail/office)</td>
</tr>
<tr>
<td>$B_{inst}$</td>
<td>Ln(institutional)</td>
</tr>
<tr>
<td>$B_{res}$</td>
<td>Ln(residential)</td>
</tr>
</tbody>
</table>

**Graph Representation:** We follow several steps to construct the graph. First, we map the signals’ locations into Open Street Map. Second, considering the signalized intersections as nodes we connect each of them with their neighboring nodes. To find the neighboring nodes, we select an origin node and find the shortest path from the origin node to all the other nodes. The nearest node on each of these shortest paths is the neighboring node to the origin. Finally, after constructing the network, we represent the network using adjacency matrix (see equation 3.1 and 3.2). We use the travel distance as weight for the graph adjacency matrix. We also perform gaussian transformation on the weighted graph adjacency matrix,

$$A(i, j) = \begin{cases} 
\exp \left( -\frac{d^2_{ij}}{\sigma^2} \right), & \text{if } i \neq j \\
0, & \text{otherwise}
\end{cases} \quad (3.5)$$

Graph Representation: We follow several steps to construct the graph. First, we map the signals’ locations into Open Street Map. Second, considering the signalized intersections as nodes we connect each of them with their neighboring nodes. To find the neighboring nodes, we select an origin node and find the shortest path from the origin node to all the other nodes. The nearest node on each of these shortest paths is the neighboring node to the origin. Finally, after constructing the network, we represent the network using adjacency matrix (see equation 3.1 and 3.2). We use the travel distance as weight for the graph adjacency matrix. We also perform gaussian transformation on the weighted graph adjacency matrix,

$$A(i, j) = \begin{cases} 
\exp \left( -\frac{d^2_{ij}}{\sigma^2} \right), & \text{if } i \neq j \\
0, & \text{otherwise}
\end{cases} \quad (3.5)$$

Graph Representation: We follow several steps to construct the graph. First, we map the signals’ locations into Open Street Map. Second, considering the signalized intersections as nodes we connect each of them with their neighboring nodes. To find the neighboring nodes, we select an origin node and find the shortest path from the origin node to all the other nodes. The nearest node on each of these shortest paths is the neighboring node to the origin. Finally, after constructing the network, we represent the network using adjacency matrix (see equation 3.1 and 3.2). We use the travel distance as weight for the graph adjacency matrix. We also perform gaussian transformation on the weighted graph adjacency matrix,

$$A(i, j) = \begin{cases} 
\exp \left( -\frac{d^2_{ij}}{\sigma^2} \right), & \text{if } i \neq j \\
0, & \text{otherwise}
\end{cases} \quad (3.5)$$

Graph Representation: We follow several steps to construct the graph. First, we map the signals’ locations into Open Street Map. Second, considering the signalized intersections as nodes we connect each of them with their neighboring nodes. To find the neighboring nodes, we select an origin node and find the shortest path from the origin node to all the other nodes. The nearest node on each of these shortest paths is the neighboring node to the origin. Finally, after constructing the network, we represent the network using adjacency matrix (see equation 3.1 and 3.2). We use the travel distance as weight for the graph adjacency matrix. We also perform gaussian transformation on the weighted graph adjacency matrix,

$$A(i, j) = \begin{cases} 
\exp \left( -\frac{d^2_{ij}}{\sigma^2} \right), & \text{if } i \neq j \\
0, & \text{otherwise}
\end{cases} \quad (3.5)$$
where, $\sigma$ indicate the standard deviation of distances.

### 3.7.2 Baseline Models

We implement two baseline models to compare the performance of the proposed GCN-LSTM and GCN Encoder Decoder model.

**LSTM:** In the LSTM model, we use single LSTM layer to predict traffic for next 4 hours. In the hidden layers we assign 196 (number of nodes) hidden neurons. The output layer is a fully connected layer with tanh activation function.

**Convolutional LSTM:** In the Convolutional LSTM (ConvLSTM) model we stacked a convolution layer with LSTM layer. Convolutional layer use convolution filter to extract the spatial correlation among traffic features in among neighboring intersections. We experiment with different size of the kernel ($k$) and find that the model performs best for a kernel size of 3. The output from the convolutional layer is feed into the LSTM layer to capture temporal correlation among traffic features while predicting traffic flow over a temporal sequence.

### 3.7.3 Model Training

We use 80% of the data for training (learning the parameters), 10% for validation (tuning hyper-parameters), and rest 10% of the data is used for testing (checking performances) the model. Based on the validation accuracy we tune the hyper-parameters such as learning rate, types of activation functions (i.e., tanh, sigmoid etc.), maximum number of iterations. Once the final model parameters are fixed, we test it on the test data set.

We train the model using mean squared error (MSE) as the loss function. At each iteration, the model estimates the MSE for the estimated traffic movement volume ($\hat{F}_{i+p}^{im}$) and the actual movement ($F_{i+p}^{im}$) volume for a given intersection ($i$), and movement types ($m$). Afterward, the gradient of the loss function is backpropagated to adjust the weights to reduce loss function value.
The loss function can be defined as:

\[
L_m = \text{Loss}(f_{t+p}^{im}, \hat{f}_{t+p}^{im})
\]

\[
MSE = \frac{1}{p \times i \times m} \sum_{p=1}^{P} \sum_{i=1}^{N} \sum_{m=1}^{M} (f_{t+p}^{im} - \hat{f}_{t+p}^{im})^2
\]

where, \( \text{Loss}(.) \) is the function to estimate the error between the actual \((f_{t+p}^{im})\) and estimated values \((\hat{f}_{t+p}^{im})\). \(i\) and \(m\) denotes the intersections (i.e., nodes) and movement types (i.e., NL (northbound left), NT (northbound through), NR (northbound right) etc.).

We implement the model using pytorch library [40] and train it in Ubuntu 18.04.5 LTS (GNU/Linux 5.4.0-62-generic x86_64) supported by a cluster of four NVIDIA RTX 2080Ti 11 GB GPUs. While training the model, we track the training and validation loss values to check whether the model is overfitting or not. From the loss values, we find that it takes about 100 epochs with a learning rate of 0.001 for the model to converge (i.e., similar train and validation loss value) (Figure 3.7). Moreover, after 100 epochs, the value of the loss function for the validation data gradually starts increasing, indicating that the model starts to overfit. We use Adaptive Moment Estimation (ADAM) to train the model. Compared to other optimizers such as Adaptive Gradient (AdaGrad), Root Mean Square Propagation (RMSProp) etc., ADAM optimizer gives more stable solutions, which means MSE values for train and validation data almost remain same after convergence.
3.7.4 Model Training

Once the final model is fixed, we test it on the test data set. We calculate Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and GEH [62] statistics as performance measures to check the accuracy of the implemented model. Performance metrics are defined as,

\[
RMSE = \sqrt{\frac{1}{p} \sum_{p=1}^{P} \frac{1}{N} \sum_{l=1}^{N} \frac{\sum_{m=1}^{M} (F_{t+p}^{im} - \hat{F}_{t+p}^{im})^2}{M}}
\]  

(3.8)

\[
MAE = \frac{1}{p} \sum_{p=1}^{P} \frac{1}{N} \sum_{l=1}^{N} \frac{\sum_{m=1}^{M} |F_{t+p}^{im} - \hat{F}_{t+p}^{im}|}{M}
\]  

(3.9)

\[
GEH = \sqrt{\frac{2(F_{t+p}^{im} - \hat{F}_{t+p}^{im})^2}{\hat{F}_{t+p}^{im} + F_{t+p}^{im}}}
\]  

(3.10)

where, \(F_{t+p}^{im}\) and \(\hat{F}_{t+p}^{im}\) indicate actual and predicted movement volumes for each intersection (i) and movement types (m) at time (t + p).

In Table 3.3, we report the performance of model on the test data. To test the sensitivity of the model over different data samples, we randomly split the data to generate 10 different train, test and validation datasets. Finally, we train 10 different models and report the mean and standard
deviation of the estimated performance measures on the test data sets. Based on performance measures, we find that the proposed GCN-LSTM model performs best compared to other baseline models. The RMSE and MAE values of the model for through traffic movement are 59.27 and 23.072, respectively. However, RMSE and MAE provide aggregate information (average over all the outputs) on the performance of the models, hence, we also estimated $R^2$ score. As shown in table 3, the $R^2$ score for all the movement types of the proposed model is above 0.98 indicating that the model can learn the traffic movement patterns very well (Figure 3.8 (a)).

Table 3.3 Comparisons among different models to predict traffic volume over 4-hour sequence for different movement types

<table>
<thead>
<tr>
<th>Model</th>
<th>RMSE</th>
<th>MAE</th>
<th>$R^2$ Score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Left</td>
<td>Through</td>
<td>Right</td>
</tr>
<tr>
<td>LSTM Mean</td>
<td>5.768</td>
<td>89.784</td>
<td>3.457</td>
</tr>
<tr>
<td>Std</td>
<td>0.343</td>
<td>6.885</td>
<td>0.191</td>
</tr>
<tr>
<td>GCN-Encoder-Decoder Mean</td>
<td>4.888</td>
<td>82.283</td>
<td>2.855</td>
</tr>
<tr>
<td>Std</td>
<td>0.258</td>
<td>3.309</td>
<td>0.141</td>
</tr>
<tr>
<td>Conv-LSTM Mean</td>
<td>4.177</td>
<td>61.145</td>
<td>2.572</td>
</tr>
<tr>
<td>Std</td>
<td>0.114</td>
<td>1.945</td>
<td>0.074</td>
</tr>
<tr>
<td>GCN-LSTM Mean</td>
<td>4.018</td>
<td>59.372</td>
<td>2.458</td>
</tr>
<tr>
<td>Std</td>
<td>0.062</td>
<td>0.706</td>
<td>0.065</td>
</tr>
</tbody>
</table>

We also estimate the absolute difference between the actual and predicted traffic volume. Figure 3.8 (b) shows the cumulative distribution functions (CDF) of absolute errors for all the movement types. For left and right turn movement, more than 95% of the predicted volumes have an absolute error less than 5, whereas for through movement about 90% of the predicted volumes are less than 60. This is because through movement volumes are higher compare to left and right turn movement.
(a) Comparison between actual and predicted traffic volume
(b) Cumulative distributions of absolute different between actual and predicted traffic volume

**Figure 3.8** Performance of the model in turning and through movement prediction
Figure 3.9 shows the correlation between actual and predicted movement volumes for peak hours only. From the figure, we find that the differences between the predicted and actual volumes are quite low (GEH < 5), which means even in case of higher demand the model performs very well to learn the traffic movement volumes over the network.

**Figure 3.9** Predicted movement volumes for all intersections at peak hours (7 am - 11am & 3pm – 8pm)

Figure 3.10 and 3.11 show the distribution of through movement types for all the intersection at a given weekday and weekend time periods (i.e. prediction four-hour temporal sequence). For a few intersections we do not have all the movement types, in those cases the model predicts the volume as zero or close to zero (<1). So, the figures include values for a valid movement type of an
intersection based on the actual data. From the figures we find that for both weekday and weekend the model captures the distribution of traffic movement volumes very well.

![Graphs showing predicted through movement volumes for all intersections for a given weekday time period (Wednesday) timestamp 08-10-2016 14:00:00 - 18:00:00)](image)

Figure 3.10 Predicted through movement volumes for all intersections for a given weekday (Wednesday) time period (timestamp 08-10-2016 14:00:00 - 18:00:00)
Figure 3.11 Predicted through movement volumes for all intersections for a given weekend (Sunday) time period (timestamp 2016-11-13 14:00:00-18:00:00)
3.8 Conclusions

Accurate traffic forecasting at a network-scale is critical to ensure proactive decision making and optimal action plans for traffic operations and management. However, traditional network models involve complex mathematical or simulation-based approach, which require higher computational time in predicting traffic at the level of an intersection including turning volume. In this chapter, we present an alternative data-driven approach adopting the concept of graph convolution. We propose two different modeling frameworks \textit{GCN-LSTM} and \textit{GCN-Encoder Decoder} to predict hourly traffic volume over multiple time steps. From the model results, we find that \textit{GCN-LSTM} model outperforms other baseline models. The overall $R^2$ value of the model is close to 1 indicating that the model captures the traffic movement volumes very well. Moreover, the absolute difference between actual and predicted volumes are quite low (GEH<5); for right turn, through and left turn movement RMSE values are 4.02, 59.37, and 2.47 respectively.
CHAPTER 4: UNDERSTANDING NETWORK WIDE HURRICANE EVACUATION TRAFFIC PATTERN FROM LARGE-SCALE TRAFFIC DETECTOR DATA

4.1 Introduction

Efficient evacuation traffic management requires detailed evacuation plan [63] including proactive measures to overcome unexpected events such as traffic incidents, a change of hurricane path causing unexpected demand surge etc. Failure to address these factors could result in potentially sub-optimal traffic operation causing severe traffic congestion and delay. For example, during hurricane Katrina, only 60% of the projected vulnerable people were able to evacuate, while during Hurricane Rita enormous response to evacuation orders created excessive traffic problems (e.g., 100-mile-long traffic jams, out of fuel etc.) and dozens of accidents or heat related deaths [64]. Optimal evacuation planning largely depends on better understanding of evacuation traffic behavior from historical data, proactive measures incorporating real-time data, and reliable models to project evacuation travel demand.

In this chapter, we present a data pipeline incorporating automated data quality checking, data imputation, and spatiotemporal visualization of network wide traffic to understand the changes in traffic patterns during hurricane evacuation. We collect Microwave Vehicle Detection System (MVDS) data for four major highways in Florida: I-75, I-95, I-4 and Florida Turnpike, these routes serve most of the evacuation traffic during Hurricane Irma and extensively analyze these data to understand the nature and extent of Hurricane Irma’s evacuation traffic and the quality

---

of these data for evacuation traffic modeling. Moreover, different factors such as time and place of evacuation order, population under evacuation order, location of the evacuees, evacuation start time, and traffic congestion are critical to predict evacuation traffic demand in future hurricanes. Since MVDS detectors are widely deployed in major highways of Florida, we perform an empirical analysis to understand the impact of these spatiotemporal factors on evacuation traffic patterns. Thus, this particular research has made several contributions for evacuation traffic analysis and modeling using large-scale traffic detector data:

- It identifies a new data source to understand evacuation traffic in real time.
- It develops a data pipeline incorporating extensive data assessment approaches to examine the quality of the network wide traffic detectors’ data including imputation techniques for evacuation traffic analysis and modeling.
- It analyzes large-scale traffic detector data during Hurricane Irma’s evacuation, providing insights on network wide spatiotemporal patterns of evacuation traffic. The data and insights obtained will help emergency agencies better understand the extent and spatiotemporal distribution of evacuation traffic.
- It assesses the influence of different spatiotemporal features on changes in evacuation traffic patterns.

The findings of this research have potential implications to deal with the challenges in large-scale network level evacuation traffic management.

4.2 Data Description

Hurricane Irma made its landfall at Florida Keys on September 10, 2017 at category 4 intensity; then it passed over several regions of Florida between September 10, 2017 and September 12, 2017. Prior to the landfall, the authorities started ordering mass evacuation from September 6,
2017 for different evacuation zones based on the location and projection of hurricane path. A massive demand surge was seen on the highways after these evacuation orders were issued. To understand major evacuation routes, we observed previous evacuations patterns, which show that a large portion of residents living in Florida evacuates to Georgia or adjacent States. Thus, four major freeways I-75, I-95, I-4 and Florida Turnpike connecting them were expected to serve a substantial amount of evacuation traffic during Hurricane Irma.

To analyze evacuation traffic patterns, we have collected traffic data for the northbound direction of I-95, I-75, Florida Turnpike, and eastbound direction of I-4 (Figure 4.1). We have collected data form Regional Integrated Transportation Information System (RITIS) from September 4, 2017 to September 9, 2017. RITIS gathers data from Microwave Vehicle Detection System (MVDS) detectors deployed by the Florida DOT, giving real-time information on traffic speed, volume, and occupancy at a very high resolution (20 to 30s frequency). For analysis purpose, we aggregate the data over 5 min interval.
Figure 4.1 Distributions of the 1426 detectors in the study network

In general, MVDS detectors have a minimum 200-feet range and the capability to detect 8 lanes of traffic. These detectors cover multidirectional traffic and most of the entry and exit ramps; the distance between two consecutive detectors varies between 0.5 and 1.5 miles. We extract information from 1426 detectors without any entry or exit ramp detectors (Figure 4.1).

4.3 Data Preprocessing

The raw data collected from traffic detectors are subjected to errors. Several factors such as detector’s malfunctioning, false encoding during storing the data into the server, overlapping of multiple entries, duplicate entries, bad weather conditions etc. can cause errors. Moreover, during congested stop and go traffic conditions, sometimes microwave radar detectors fail to detect vehicles, hence provide misleading information. Therefore, before proceeding to any data
analysis, we need an extensive data cleaning and quality checking. Figure 4.2 shows the framework for the data processing steps.

![Figure 4.2 Framework for data processing](image)

To check the quality of the data, we followed several steps starting with checking the percentage of missing data for different detectors. We consider the detectors having a higher percentage of missing values as unreliable ones. Moreover, data imputation is not feasible for the detectors with too many missing values as it will produce unrealistic data distribution. Considering this issue, we retain the detectors having missing values less than 20% of total data samples. Moreover, we find a few detectors with higher percentage of zero entries. Considering the regular traffic pattern on freeways, we expect near zero or minimal traffic flow during nighttime, especially from 10 pm to 5 am, which means about 25% of the total entries per day per detector could be zero. However, during evacuation, mostly from September 06 – September 08, 2017, we may not see this trend, there will be a large number of vehicles throughout the day regardless of the time. Additionally, since Hurricane Irma made its landfall on September 10, we expect minimal traffic on September 9, 2017. Thus, it is likely that 30-40% data for each detector might have zero entries. However,
we observe that some detectors have between 40% and 100% zero values, which indicates anomaly in vehicle detection. Therefore, we remove the detectors with more than 40% zero entries.

We also consider the lane capacity of the highways. In regular periods under free flow condition, the capacity of a highway segment varies from 2,000 to 2,400 vehicle per hour per lane (vphpl) depending on the prevailing traffic speed. However, during evacuation there could be a significant reduction in traffic speed. We analyzed the traffic speed for I-75 during Hurricane Irma’s evacuation and found average traffic speed varying from 50 mph to 65 mph. Thus, overall capacity of the highways (2,000 vphpl) should be less than theoretical values. However, we cannot directly estimate actual highway capacity, it depends on many factors such as incidents, traffic crashes, lane closures etc. which can substantially reduce capacity. On the other hand, emergency shoulders use (ESU) during evacuation can increase overall capacity. During Hurricane Irma’s evacuation ESUs were used from September 7- September 9, 2017 to help evacuating traffic along I-75. Such ESUs increase the overall throughput of the roadways by 25% [65]. Thus, we consider the maximum possible capacity of the highway as 2500 vphpl. For each detector data sample, we divide the total traffic volume per hour by the number of lanes to obtain vehicles per hour per lane (vphpl), these values should remain close to the maximum capacity (2500 vphpl).
Figure 4.3 Distribution of data samples based on vehicles per hour per lane

Figure 4.3 shows the distribution of vphpl values combining all the detectors’ data. Almost all the data samples except a few, have vphpl less than 2500. A few data samples show unrealistic values, nearly 25,000 vphpl. We check all the detectors individually and find that, for most of the detectors, less than 5% of data samples show higher values of vphpl compared to per lanes capacity. For each detector, the erroneous volume (corresponding to vphpl > 2500) is replaced with “NaN” values.

Later, we use a technique known as multivariate iterative imputation adapting Bayesian ridge regression as estimator to impute the missing or “NaN” values. To fit the estimator, we use time of the day (hour), day of the week (day), and volume with missing values as inputs. For each imputation the algorithm takes a sample from gaussian posterior of the fitted estimator. We use Python scikit learn [60] library to implement the algorithm. The details about the data imputation algorithm are provided in reference [59].
4.4 Data Exploration

In this section, we determine the size of the population under mandatory evacuation orders for different evacuation zones to understand the impact of evacuation order on traffic demand. Previous studies found that households are more likely to evacuate under a mandatory evacuation order; hence the spatiotemporal traffic patterns are likely to depend on the timing of evacuation declaration and type of evacuation order.

4.4.1 Spatiotemporal Pattern of Population Under Evacuation

We collect the time and location of evacuation orders issued for different areas for Hurricane Irma from the Florida Division of Emergency Management. However, the declaration dates of evacuation order for all the zones are not available in a single source, thereby, in a few cases, we collect the declaration date by manually checking the emergency management agency’s social media posts (e.g., Twitter, Facebook) of the respective county and contemporary news article available online. Figure 4.4 shows the mandatory evacuation zones with declaration time. We observe that most of the evacuation zones are by the coast; smaller zones in the central part of Florida mainly represent mobile homes or low-lying areas vulnerable to inland flooding.
Figure 4.4 (a) Zone wise mandatory evacuation orders with corresponding declaration dates (b) Population of mandatory evacuation zones

Florida Keys and other low-lying zones such as Everglades were issued mandatory order in early September 5, 2017. Evacuation zones in the east coast, such as Miami-Dade, Daytona were issued evacuation order on September 7, 2017 (Hurricane Irma was supposed to hit the east coast of Florida until Sep. 7, 2017). After September 7, 2017, as the projected path shifted from the east coast to the west coast, evacuation zones of Naples, Cape Corals, Tampa, Levy, Jacksonville, were ordered mandatory evacuation on September 7, 2017 and onward (see Figure 4.4(a)). We have collected population data for the mandatory evacuation zones to understand how many people were under mandatory evacuation order. Since, population data is not available for the evacuation zones, we collect block group level population data from 2017 5-year American Community Survey and sum the population that falls within an evacuation zone to retrieve the population for the zone. The light grey boundaries of Figure 4.4(a) and 4.4(b) represent the block group boundary within Florida. Spatially most people under mandatory evacuation are from Miami, Fort Meyer, and Tampa area during Hurricane Irma (see Figure 4.4(b)); the highest number of people were under a mandatory evacuation order on Sept. 8, 2017 (about 3,420,271 people), followed by Sept. 9, 2017 (about 2,629,161 people).
### 4.4.2 Spatiotemporal Patterns of Evacuation Traffic

In a normal operating condition, traffic state shows predictable patterns such as heavy traffic demand during peak hours (e.g., 4pm to 8pm) and comparatively lighter traffic demand during off peak hours (e.g., 8pm to 12 am). However, during an emergency event such as hurricane evacuation, overall traffic condition has to bear severe disruption due to a drastic increase in traffic demand [56]. To compare the evacuation traffic volume with non-evacuation period traffic volume, we have collected traffic data from May 1 to August 31, 2017. For each detector, we calculate the average hourly traffic volume for different time periods (1 to 24 hr.) and days (weekdays and weekends). Finally, we calculate the difference between hourly traffic volume during evacuation period and non-evacuation period. Figure 4.5 shows the difference between evacuation traffic and regular traffic for I-75 and I-95. From the figure, we find that during Hurricane Irma’s evacuation from September 6 to September 8, 2017, overall traffic flow is higher all the time regardless of whether it is a peak hour or not; there is a significant amount of traffic congestion on the major interstates even after peak hours (4pm - 8pm).

We also explore the spatial patterns of evacuation traffic to understand the impact of mandatory evacuation orders on traffic demand variations (Figure 4.6). During Hurricane Irma, mandatory evacuation order was placed at different evacuation zones of Florida from September 5, 2017 to onwards depending on the predicted time and intensity of hurricane landfall. Initially the projected path showed south east coastal region of Florida (Miami, West Palm, Fort Lauderdale etc.) as the most critical zones: these areas were supposed to take major impact from Hurricane Irma. From September 5 - September 7, 2017 nearly 3.5 million people were under mandatory evacuation from these regions including some major cities such as Miami, West Palm Beach, Fort Lauderdale etc. and other low-lying areas such as Florida Keys, Everglades. Thus, from September 6, 2017
to September 7, 2017, we observe a drastic increase in traffic demand on I75 and I95, two major highways connecting south east and south west regions with the rest of Florida and the neighboring states such as Georgia and Alabama. However, since after September 7, 2017 Irma’s projected path shifted from the east coast to the west coast, many cities such as Naples, Cape Corals, Tampa, Levy, Jacksonville became the most vulnerable area. Nearly 6 million people were under mandatory evacuation from these regions. The added traffic from these population caused a severe congestion on downstream of I75 and I95. To serve this increased demand, emergency shoulder use (ESU) was activated in I-75 on September 7, 2017 from Ocala towards Georgia. On Sept 9, 2017 before the hurricane landfall day we observe a large volume of traffic at downstream of I-75 and I-95 (Figure 4.6), while in the upstream zones traffic volumes were nearly zero. Traffic volumes started to decrease after 1pm on September 9.

One of the major challenges in evacuation traffic management is to deal with such unexpected events, especially when there is severe traffic congestion at the last moment close to hurricane landfall. To make thing worst, all the shelters may be full. Another important issue with evacuation management is late response of the evacuees to evacuation orders. Most of the cases people wait for the last moment before taking evacuation decisions, thus causing severe traffic congestion at the eleventh hours. To deal with such unexpected events, evacuation plans should learn from the experiences of previous hurricanes and prepare accordingly.
Figure 4.5 Spatiotemporal traffic flow distribution, here we plot the traffic detectors based on (latitudes, longitudes)/unique zone ids
4.5 Results

In this section, we demonstrate the application of such network-wide traffic data by estimating the influence of different spatiotemporal factors on variations of evacuation traffic patterns.

4.5.1 Linear Regression

We implement a simple linear regression model to estimate the influences of different factors on spatiotemporal variations of evacuation traffic flow. We consider several exogenous variables such as whether the detector located in an evacuation zone or not, distance of the detector from nearest evacuation zone, county wise population with respect to detector location and time period of the day (i.e., Late Night, Early Morning, Morning, Noon, Evening, Night), hour left before hurricane landfall and cumulative total of population under mandatory evacuation from September 5 to onwards aggregated over one-hour period. We also use average traffic volume for each of the detectors over different hour (1 -24 hr.) and different days (weekdays and weekends) to capture the usual traffic demand for different locations over the transportation network, it will capture the importance of different detectors (or location) based on its position inside the network.
Table 4.1 reports all the significant variables based on t-stat. The variables are significant at 95% confidence interval (P-value < 0.05).

Table 4.1 Estimates from linear regression model.

<table>
<thead>
<tr>
<th>Variables</th>
<th>Estimates</th>
<th>t-stat</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>311.97</td>
<td>23.79</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>Average flow in non-evacuation condition</td>
<td>0.59</td>
<td>263.64</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>Distance from the nearest evacuation zone</td>
<td>-4.90</td>
<td>-17.90</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>Population (in 10,000) under mandatory evacuation (shifted by 18 hr)</td>
<td>0.24</td>
<td>11.39</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>Hours before landfall</td>
<td>7.37</td>
<td>80.40</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>Early Morning</td>
<td>292.75</td>
<td>22.93</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>Morning</td>
<td>302.09</td>
<td>22.22</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>Noon</td>
<td>338.64</td>
<td>25.42</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>Evening</td>
<td>167.07</td>
<td>11.69</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>Night</td>
<td>-116.21</td>
<td>-9.41</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>Late Night (reference)</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

In the table, the variable “Average flow in non-evacuation condition” indicates the average hourly traffic volume (i.e., 1 to 24 hr.) for different locations on the network and different days of the week (i.e., Mon., Tues. etc.) over a period of May 1 to August 31, 2017. From the model estimates, we find that the coefficient corresponding to this variable is positive, which means that during evacuation period traffic flow will be higher at the locations with high traffic volume in non-evacuation periods. The reason is that roadway segments with high capacity accommodates more traffic hence overall traffic flow will be higher regardless of evacuation or
non-evacuation period. Additionally, some of the roadways connect highly populated major cities (i.e., Miami, Tampa, Orlando etc.), consequently they serve heavy traffic demand during non-evacuation period. Likewise, they have to accommodate large volume of traffic during evacuation period due to mass evacuation from these populated cities.

The coefficient corresponding to the variable “distance from the nearest evacuation zone” is negative, indicating that if a detector is located far away from evacuation zone it will be less impacted by evacuation traffic. We observe that the central part of the interstate network (I-4) is far away from evacuation zones, they are less impacted from evacuation traffic. In case of the variable “population under mandatory evacuation” we consider the time lag between declaration of evacuation order and the time when people start to evacuate. To capture this time lag we continuously shift the cumulative population by one \((t+1)\) hour interval and estimate the coefficient of that variable by running the model. We do not see any significant value of associated coefficient (estimated value remains negative) till 17 hr., however after that time period which means from 18 hour to onwards, we observe that the coefficient associated with the variable total population under mandatory order is positive; moreover, it gradually increases with the increase in lag time (from 18 to onwards). It means that after the declaration of evacuation order it takes approximately 18 hr. for the people to start evacuating, thereby increase total traffic on the roadway. We also find that, the coefficient associated with variable “hours before hurricane landfall” is positive, which means that closer to landfall time the chances of evacuation is lower, as a result total traffic flow during this time period will also remain low. From the spatiotemporal analysis, we observe similar patterns. In the first three days of evacuation period, traffic is significantly higher, whereas close to landfall time traffic flow is lower except some downstream
location of the network. Moreover, from the estimated results we find that people are less likely to evacuate during nighttime.

**4.5.2 Tree Based Model to Estimate the Feature Importance**

We further analyze the importance of these features running tree-based models such as decision tree, random forest, gradient boosting and extreme gradient boosting regression models. Figure 4.7 (a) demonstrates the importance of all the significant features running different models. From the figure we find that two features: average traffic volume for a given detector under regular condition and hours left before hurricane landfall are the two most important variables when predicting evacuation traffic.

![Feature Importance for Tree Based Models](image1)

**Figure 4.7 Illustrates** (a) features importance (b) Root Mean Square Error (RMSE) and (c) $R^2$ Score for the tree-based models

$xgb = \text{extreme gradient boosting; gbr = gradient boosting; rnf = random forest; tree = decision Tree}$
We also check the performance of the models for evacuation traffic prediction considering two performance measures: root mean square error (RMSE) and $R^2$ score. As shown in Figure 4.7(b) and (c), extreme gradient boosting algorithm performed best (RMSE = 450) compared to other three models. Although we do not consider any information on individual evacuation decision, the model performed reasonably well indicating the importance of these features in developing evacuation traffic prediction model.

4.6 Conclusions

In this chapter, we present an extensive analysis to understand the coverage and quality of MVDS detectors’ data (i.e., traffic volume) for evacuation traffic analysis and modeling for hurricane Irma. We have made some reasonable assumptions in this study to clean the data. Traffic agencies can make different assumptions about the threshold values. However, we believe that these assumptions will not significantly impact the amount of data to be available for understanding hurricane evacuation patterns.

We conduct spatiotemporal data analysis to understand the changes in evacuation traffic pattern during Hurricane Irma. The analysis reveals that although evacuation order was placed early (September 6, 2017), still a significant traffic congestion occurred at the downstream of I-75 and I-95 just before the landfall day. This happened because of the changes in hurricane path from the east coast to the west coast, forcing more people from south Florida and Jacksonville to evacuate at the last moment. The empirical analysis further reveals that there is at least about an 18-hr. time lag between the time of evacuation order and the time when people started to evacuate. Moreover, the location of the detectors with respect to evacuation zone, time left before hurricane landfall and time period of the days influence the variations of evacuation traffic. Such findings have potential implications in large network-scale evacuation traffic modeling.
Evacuation traffic management is a complex process that requires rapid responses from emergency management agencies to address unexpected events. Hence, it is important that transportation agencies utilize available real-time data sources to understand the responses from evacuees as evacuation unfolds over time. This study serves this purpose by identifying a new data source, providing valuable insights on the quality of the data, and their capacity to model spatiotemporal correlation among traffic state to predict network level evacuation traffic demand at different zones.
CHAPTER 5: A DEEP LEARNING APPROACH FOR NETWORK-WIDE DYNAMIC TRAFFIC PREDICTION DURING HURRICANE EVACUATION

5.1 Introduction

Hurricanes have become more intense and frequent due to climate change and other related reasons. As a result, coastal residents of the United States are becoming more vulnerable to the impact of hurricanes. To save lives and reduce suffering of people during such events, emergency management agencies need efficient and pro-active strategies to ensure timely evacuation. Over the past decades, several traffic management strategies [23] such as emergency shoulder use, contraflow operations, traffic signal control, route guidance etc. have been proposed to alleviate congestion during evacuation. However, a proactive deployment of these measures requires a better understanding of prevailing traffic conditions and prediction of future traffic for long-term horizon (> 1 hour). Thus, one of the critical aspects of evacuation traffic management is to understand the spatial and temporal distribution of congestion and to reliably predict future traffic condition. Based on this information, emergency traffic managers can decide where, when, and for how long a strategy should be deployed to reduce delays.

This chapter presents a novel deep learning architecture to overcome these challenges for evacuation traffic prediction. First, we develop a dynamic graph convolutional LSTM (DGCN-LSTM) model with contextual understanding of congestion propagation inside the network. To overcome the limitation of data scarcity and model overfitting issue, we train the model on regular period traffic data collected from May 2017 to August 2017. Later, we transfer this model to evacuation period with additional neural network block as a controller to predict traffic during

---

evacuation period. The controller uses evacuation travel demand features such as distance from the evacuation zone, time to landfall, and other zonal level features to control which information from the *DGCN-LSTM* to be included to predict traffic during evacuation period. So, the final evacuation prediction model has the contextual understanding of the congestion propagation inside the network; at the same time, it can discard any unnecessary information such as seasonality in traffic patterns which might be out of context during an evacuation period. This particular research has several contributions for evacuation traffic prediction using large-scale traffic detector data:

- It identifies the potential challenges to deal with large-scale real-time traffic data for evacuation traffic analysis and prediction.
- It develops a dynamic graph convolutional neural network model to learn the congestion propagation for a large-scale network.
- It implements a transfer learning based deep learning architecture to incorporate the contextual understanding of congestion propagation inside a network for evacuation traffic prediction.

### 5.2 Literature Review

Evacuation traffic management can benefit from how accurately we can predict traffic in real time. However, practices in evacuation traffic management mostly focus on behavioral analysis of evacuees to understand/predict evacuation decisions [23, 66–69]. Previous studies investigated evacuation behavior during hurricanes focusing on understanding the factors relating to evacuation decisions [70–74], mobilization time [69], departure time [75, 76], and destination choice [77, 78]. Although a vast number of research has been carried out on individual-level evacuation decision-making [78, 79] to determine evacuation traffic demand, these approaches depend on surveys data that are expensive and difficult to conduct as a hurricane unfolds in real-time. Although a few studies has explored evacuation traffic patterns, these are limited to
understanding operational capacity loss of highways during a hurricane evacuation [64, 80–82]. However, these studies provide several key insights on challenges in evacuation traffic modeling.

One of the major challenges in modeling evacuation traffic is that traffic patterns during evacuation significantly differ from non-evacuation period, characterized by higher demand and irregular traffic variations [83]. Evacuation traffic demand is more uncertain due to sudden changes in hurricane path [84]. To model such uncertain demand variations, we need a demand responsive model. Few studies [22–25] have adopted mathematical modeling and simulation-based approaches to model evacuation traffic. However, such approaches rely on certain assumptions to estimate the demand for evacuation traffic which might fail to present the actual evacuation scenario.

A data-driven method can be a viable solution to overcome this issue. A data-driven approach predicts the traffic state by finding the current traffic state from historical traffic patterns, without any assumption on user behavior [85], making the predictions more robust and demand responsive. Data-driven methods have been applied in numerous predictive modeling such as traffic speed prediction [55, 86, 87], travel time prediction [88], traffic flow prediction [89–91] etc. In addition to regular traffic condition, studies have also explored the application of data-driven methods for traffic management during a major event such as hurricane evacuation [56] and traffic incidents [92].

Existing data-driven approaches have two major limitations: (i) they do not consider travel demand variations while predicting future traffic; (ii) they only predict short-term traffic states (speed, flow, travel time). In case of evacuation traffic, it is critical to predict traffic well ahead of time (e.g., >1 hr.) to provide transportation agencies enough time to deploy traffic management strategies (i.e., signal control, emergency shoulder use etc.). However, to accurately predict traffic
for a longer time period (>1 hr.) we need to account for demand variations over different time periods. In this study, we develop a data-driven method to predict future time series of traffic flows by learning the historical data from roadway detectors. We develop a deep learning model combining evacuation traffic demand related features with regular traffic features to predict evacuation traffic flow for a long-term horizon (1 to 6 hours).

5.3 Problem Formulation

To implement the method, we construct a network of traffic detectors where each detector indicates a node. In this network travel time between two nodes dynamically changes over time. To capture the dynamics, we define the network as a dynamic graph \( G_t(v, E, A_t) \) where \( v \) denotes the set of nodes (i.e., detectors) and \( E \) denotes the set of links between nodes \((i, j)\). \( A_t \) represents the connectivity between nodes as a weighted adjacency matrix, where weights are based on travel time between any two nodes \((i, j)\), defined as follows:

\[
A_t(i, j) = \begin{cases} 
  t_{t_i, j}^t, & \text{if } i \neq j \\
  0, & \text{otherwise}
\end{cases}
\]  

(5.1)

where, \( t_{t_i, j}^t \) denotes the travel time between the nodes \( i \) and \( j \) at time \( t \). The connectivity inside an adjacency matrix detects which neighboring nodes \( (j) \) will be influenced by the traffic condition at a given node \((i)\). Moreover, in a time series problem the existing traffic condition at a given node \((i)\) will also influence its future traffic condition, which means each node is temporally self-influenced. This is represented by adding an identity \((I)\) matrix with the adjacency matrix which ensures that nodes are self-accessible,

\[
A_t = A_t + I
\]

(5.2)

We aim to learn traffic flow patterns in a transportation network over multiple time steps (i.e., future time series) based on capturing the influence of congestion propagation (i.e., travel time variations) on spatiotemporal cross correlation among nodes’ traffic condition. In this
problem, traffic condition is represented as a function of traffic demand related features. Thus, we feed the model with the information on two aspects: (i) a dynamic graph indicating the variations in travel time and (ii) node level features related to traffic demand. Let, $X_t$ be the input features and $G_t(v,E,A_t)$ is a dynamic graph with weighted adjacency matrix $A_t$. The problem is defined as to learn a function $F(.)$ that maps $l$ instances of input sequence $([X_{t-l},X_{t-l+1},...,X_t])$ to predict $p$ instances of flow $(F_{t+1},F_{t+2},...,F_{t+p})$ for the entire network. Mathematically, the problem is defined as follows:

$$F([X_{t-1},X_{t-1+1},...,X_t], [G_{t-l}(v,E,A_{t-l}),G_{t-l+1}(v,E,A_{t-l+1}),...,G_{t}(v,E,A_{t})]) = [F_{t+1},F_{t+2},...,F_{t+p}]$$  \hspace{1cm} (5.3)$$

where, $l(=0,1,2,...,l)$ and $p(=1,2,3,...,p)$ indicate the input and output sequence, respectively; $X_t$ indicates traffic related features (i.e. volumes, time periods etc.) at time $t$; and the vector $F_{t+p}$ indicates the link flows for each link of the network at time $(t + p)$. We have added the description of the notations associated with the model development in Table 5.1.
Table 5.1 Description of the notation associated with the model development

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{G}$</td>
<td>Transportation network</td>
</tr>
<tr>
<td>$v$</td>
<td>Set of nodes in $\mathcal{G}$ with size of $</td>
</tr>
<tr>
<td>$\mathcal{E}$</td>
<td>Set of links in $\mathcal{G}$ with size of $</td>
</tr>
<tr>
<td>$A_t \in \mathbb{R}^{N \times N}$</td>
<td>Weighted adjacency matrix of $\mathcal{G}$, defined by Equation (1)</td>
</tr>
<tr>
<td>$I \in \mathbb{R}^{N \times N}$</td>
<td>Identity matrix</td>
</tr>
<tr>
<td>$\bar{A}_t \in \mathbb{R}^{N \times N}$</td>
<td>Neighborhood matrix defined by Equation (2)</td>
</tr>
<tr>
<td>$\bar{D}_t \in \mathbb{R}^{N \times N}$</td>
<td>Degree matrix of $\mathcal{G}$, a diagonal matrix where diagonal elements $(i, i)$ indicate the number of links coming out from a node</td>
</tr>
<tr>
<td>$t_{ij}$</td>
<td>Travel time between nodes $i$ and $j$</td>
</tr>
<tr>
<td>$l$</td>
<td>Input time sequence length (0,1, ..., $l$)</td>
</tr>
<tr>
<td>$c$</td>
<td>Number of features at each node</td>
</tr>
<tr>
<td>$X_t^{reg} \in \mathbb{R}^{N \times c}$</td>
<td>Contains all the traffic features (i.e. volumes, time periods etc.) associated with each node ($i$) of the network for regular condition</td>
</tr>
<tr>
<td>$X_t^{evc} \in \mathbb{R}^{N \times c}$</td>
<td>Contains all the traffic features (i.e. volumes, time periods etc.) associated with each node ($i$) of the network for evacuation condition</td>
</tr>
<tr>
<td>$X_t^{eD} \in \mathbb{R}^{N \times c}$</td>
<td>Contains the features related to evacuation travel demand (i.e population under mandatory, evacuation zones’ location etc.) associated with each node ($i$) of the network</td>
</tr>
<tr>
<td>$g_t$</td>
<td>Graph Convolutional filter to learn the congestion propagation inside the network</td>
</tr>
<tr>
<td>$f(.)$</td>
<td>Activation function</td>
</tr>
<tr>
<td>$W_{gc} \in \mathbb{R}^{N \times N}$</td>
<td>Learnable parameters for the convolution filter</td>
</tr>
<tr>
<td>$H$</td>
<td>Indicates the outputs from different layers of the proposed neural network architecture</td>
</tr>
<tr>
<td>$p$</td>
<td>Prediction horizon (1, ..., $p$)</td>
</tr>
<tr>
<td>$F_{t+p} \in \mathbb{R}^E$</td>
<td>Flow vector contains flows for each link (segment) of the network for the prediction horizon</td>
</tr>
</tbody>
</table>

All the bold letters denote a matrix
5.4 Methodology

5.4.1 Learning Traffic Flow Dynamics of the Transportation Network

Traffic flow dynamics in a transportation network can be represented as a flow propagation process—traffic traversing from the origin node to the destination node via neighboring nodes. That is why, traffic condition of a given node influence the traffic condition of the neighboring nodes, in other words there exists a spatial correlation among these nodes. However, at any time step whether the traffic at given a node will reach any neighboring node or not depends on the travel time between these nodes, which changes over time. So, to model the traffic flow dynamics we need to represent travel time variations of the network and utilize this information while capturing spatial correlation among the nodes.

In this study, we develop a graph convolution based deep neural network architecture to capture spatiotemporal correlation among node level traffic features for predicting traffic flows. The model has two layers (see Figure 5.1): in the first layer, we apply a graph convolution operation to capture the spatial correlation among neighboring nodes. In this approach, we derive a graph convolutional filter from adjacency matrix which represents the travel time variations of the network, thereby detects which neighboring nodes are within the shortest path distance of the origin node at a given time step. To derive the convolutional filter, we adopt a graph theoretic approach where a graph adjacency matrix is decomposed into its eigenvalues to represent the structural properties of the graph such as the strength of a node (i.e., node level features), shortest path between two nodes etc. Such a representation, when fed into a deep learning model, suffers from exploding or vanishing gradient problem due to sparsity in eigen values’ distribution. To overcome this exploding or vanishing gradient problem, Kipf and Welling [26] proposed a normalization technique to represent a graph and its intrinsic dynamics. We adopt a similar
approach and define the graph as a symmetrically normalized adjacency matrix \( (\frac{1}{2^2}A_tD_t^{-\frac{1}{2}}) \).

However, in previous applications the networks are static, hence normalized adjacency is fixed; while in our case the network is dynamic, hence the normalized adjacency matrix will change over time.

\[
g_c = (W_{gc} \circ (\frac{1}{2^2}A_tD_t^{-\frac{1}{2}}))X_t
\]

(5.4)

where, \( g_c \) indicates the convoluted feature matrix and \( W_{gc} \) indicates the parameters for the convolution filter. The convoluted feature matrix represents the state transition of the network, in other words how congestion is propagating inside the network and influencing neighboring nodes.

In the second layer, we apply a LSTM [54] model to map this convoluted feature matrix into traffic flows. The LSTM model captures the temporal dependency among traffic features.

**Figure 5.1 A dynamic graph learning approach for network-wide traffic prediction**

The main function of the graph convolution layer is to capture the variations of spatial cross correlation among nodes due to network wide travel time variations. We use the normalized graph adjacency matrix as a convolution filter and perform the convolution with node level traffic demand related features of the network. The convolution operation can be defined as follows:
while predicting traffic flows over multiple time steps (i.e., future time series). The proposed dynamic graph-based LSTM model (DGCN-LSTM) model can be defined as follows:

$$F_{t+p} = LSTM(f(gc_t)) = LSTM(f(W_{gc} \odot (D_t^{1/2}A_tD_t^{1/2}))X_t)$$

(5.5)

where, $f$ indicates the nonlinear activation function; we use rectified linear unit (relu) as an activation function.

Training such graph-based models over a transportation network requires a substantial amount of data. However, evacuations usually take place for 2 to 5 days before the hurricane landfall. When a model is trained with a small sample it will cause the model to overfit. To overcome this problem, we develop a new modeling method adopting a transfer learning technique[93]. We first train the DGCN-LSTM model over regular traffic data ($X_t = X_t^{reg}$) and later transfer this model to evacuation period. The following section describe the methods to implement the model for evacuation traffic prediction.

### 5.4.2 Network-wide Evacuation Traffic Prediction

Evacuation traffic depends on many factors such as zonal level population under mandatory evacuation, hours left before landfall, distance of a detector from the nearest evacuation zone, and different time periods of the day etc. [84]. Using these features as inputs, we can develop a simple time series-based model to predict evacuation traffic flow. However, such a model will perform poorly since, it cannot capture the spatiotemporal dependency of traffic variables as well as does not have any information on the underlying contexts of congestion propagation inside the network.

To overcome this problem, we adopt a transfer learning approach to transfer the context of network dynamics over multiple time steps (temporal sequences). However, traffic demands during evacuation significantly differ from non-evacuation condition. For example, evacuation traffic demand is higher than non-evacuation period and does not follow any regular pattern. Thus, when
applying the transfer learning approach, we need to transfer only the information relevant for an evacuation period such as information of network connection and the function how traffic flows from an upstream to a downstream location.

Figure 5.2 A deep learning architecture for evacuation traffic prediction

We develop a deep learning architecture which controls the information flow from regular traffic condition to evacuation traffic condition. The proposed deep learning architecture has four components (see Figure 5.2). The first component is the pretrained DGCN-LSTM model, we apply this model using the traffic features (i.e. volumes, time periods etc.) from evacuation period ($X_t^{evc}$) to predict traffic flow over multiple timesteps.

$$h_t^{evc} = DGCN\text{LSTM}(X_t^{evc})$$  \hspace{1cm} (5.6)

where, $h_t^{evc}$ indicates the outputs from the DGCN-LSTM model. We add this information with other evacuation traffic demand information. The second component is an LSTM layer, we apply this model to capture the temporal correlation among evacuation traffic demand related features ($X_t^{ed}$).
\[ h_{evc} = LSTM(X_t^{eD}) \]  

where, \( h_{evc} \) indicates the output from the LSTM layer. The third component is the control layer; in this layer, we define a neural network with sigmoid activation function to remove irrelevant information from the DGCN-LSTM model.

\[ f_{control} = \sigma(W_c X_t^{eD} + b_c) \]  

where, \( f_{control} \) indicates the output from the control layers which are distributed in between 0 to 1. The fourth and final component is the output layer which adds the network dynamics related information with evacuation demand to generate the final traffic prediction.

\[ F_{t+p} = f_{control} \otimes h'_{evc} + \text{tanh}(h_{evc}) \]  

In this layer, we perform an elementwise matrix multiplication between \( f_{control} \) and \( h'_{evc} \), thus some of the information will be erased prior to adding with evacuation demand. Since we assign weight \( W_c \) in the control layer, when training the model for evacuation traffic prediction it automatically learns to control the information flow from non-evacuation condition to evacuation condition.

### 5.5 Data Collection and Preprocessing

#### 5.5.1 Traffic Detector Data

To test the model, we consider a network consists of interstates highways. One of our objectives is to apply the model to predict evacuation traffic. Hence, to select the network we have observed previous evacuations patterns, which show that a large portion of residents living in Florida evacuates to Georgia or adjacent States. Thus, two major highways I-75, I95 and other two highways I-4 and Florida Turnpike connecting them are expected to serve a substantial amount of evacuation traffic during Hurricane Irma. We choose the northbound direction of I-95, I-75, Florida Turnpike, and eastbound direction of I-4 (Figure 5.3) to formulate the network.
Figure 5.3 Network of inter-state highways

We have collected data from Regional Integrated Transportation Information System (RITIS) [94] from September 4, 2017 to September 9, 2017 which covers the evacuation period of Hurricane Irma. We have also collected non-evacuation period traffic volume from May 1 to August 31, 2017. RITIS gathers data from Microwave Vehicle Detection System (MVDS) detectors deployed by the Florida DOT, giving real-time information on traffic speed, volume, and occupancy at a very high resolution (20 to 30s frequency).

The raw data collected from traffic detectors are subjected to errors. Several factors such as detector’s malfunctioning, false encoding during storing the data into the server, overlapping of multiple entries, duplicate entries, bad weather conditions etc. can cause errors. Moreover, during congested stop and go traffic conditions, sometimes microwave radar detectors fail to detect vehicles, hence provide misleading information. Therefore, before proceeding to any data analysis,
we need an extensive data cleaning and quality checking. Figure 5.4 shows the framework for the data processing steps.

**Figure 5.4 Framework for data processing**

We follow several steps to process the data for analysis. Firstly, we remove the detectors having higher percentages of missing values (>20%); secondly, we detect the outliers based on the capacity of the highway (2500 vehicle per hour per lane); finally, replace the outliers and available missing values using multivariate iterative data imputation technique [59, 60]. The details about data preprocessing is provided in chapter 4.

**5.5.2 Zonal Level Mandatory Evacuation**

We have collected the time and location of evacuation orders issued for different areas for Hurricane Irma from the Florida Division of Emergency Management. However, the declaration dates of evacuation order for all the zones are not available in a single source, thereby, in a few cases, we collect the declaration date by manually checking the emergency management agency’s
social media posts (e.g., Twitter, Facebook) of the respective county and contemporary news article available online. Figure 5.5 shows the mandatory evacuation zones with declaration time.

From the figure we find that, Florida Keys and other low-lying zones such as Everglades were issued mandatory order in early September 5, 2017. Evacuation zones in the east coast, such as Miami-Dade, Daytona were issued evacuation order on September 7, 2017 (Hurricane Irma was supposed to hit the east coast of Florida until Sep. 7, 2017). After September 7, 2017, as the projected path shifted from east coast to west coast, evacuation zones of Naples, Cape Corals, Tampa, Levy, Jacksonville, were ordered mandatory evacuation on September 7, 2017 and onward (see Figure 5.5). We have collected population data for the mandatory evacuation zones to understand how many people were under mandatory evacuation order. Since, population data is not available for the evacuation zones, we collect block group level population data from 2017 5-year American Community Survey and sum the population that falls within an evacuation zone to retrieve the population for the zone. The highest number of people were under a mandatory evacuation order on Sept. 8, 2017 (about 3,420,271 people), followed by Sept. 9, 2017 (about 2,629,161 people).
5.6 Experiments

5.6.1 Feature Extraction and Graph Representation

We followed several steps to extract the spatiotemporal features from the collected data. We prepare two types of data samples: (i) traffic data samples for regular period ($X_t^{reg}$) and evacuation period ($X_t^{evac}$); (ii) evacuation demand features ($X_t^{ed}$). All the input features are listed in Table 5.2.

Traffic data samples ($X_t$): We aggregate the traffic data for 1-hour intervals estimating traffic flow and average traffic speed. To capture the periodic nature of traffic flow variations, we group the hours into 6 different time periods such as late night, early morning, morning, noon, evening, and night. We represent these features using one hot encoding, which means each of the six time periods is represented as an indicator variable (0,1). We also extract different features to present traffic flow variations over previous day ($t_{d-1}$) and previous time period ($t_{prd-1}$).
corresponding to current day \((t_d)\) and time periods \((t_{prd})\) at time \((t)\). The extracted features include previous day and time periods’ mean and standard deviation of traffic flow). Since we do not have any data to indicate the characteristics of different zones (e.g., built environment characteristics, zonal level population etc.), we use a categorical indicator variable named “Zone ID” to represent zonal characteristics specific to the location of each detector. This variable also represents the ordering of the output sequence (i.e. 1 to 806) for all the detectors.

We formulate the traffic data sample as \([\text{number of samples } (n), \text{ input time sequence } (l), \text{ number of nodes } (N), \text{ input features } (c)]\). Since we have collected the data from 806 detectors, the number of nodes, \(N = 806\). We select 6-hour input data sequence to predict traffic for next 6 hour, so input time sequence length, \(l = 6\) and prediction horizon length, \(p = 6\). In total we have \(c = 12\) input features: Zone ID, Late Night (12am-4am), Early Morning (4am-8am), Morning (8am -12 am), Noon (12 pm-4pm), Evening(4pm-8pm), Night (8pm -12am), previous day mean traffic flow; previous day standard deviation of traffic flow, previous time period mean traffic flow, previous time period standard deviation in traffic flow. For non-evacuation period, we have 2148 data samples and for evacuation period we have 120 data samples. So, for evacuation and non-evacuation periods the input data has the shape as \([120, 6, 806, 12]\) and \([2148, 6, 806, 12]\), respectively and the target data has the shape as \([120, 6, 806]\) and \([2148, 6, 806]\), respectively.
Table 5.2 Description of the input features

<table>
<thead>
<tr>
<th>Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Z$</td>
<td>Zone ID: represent zonal characteristics specific to the location of each detector (i.e. 1,2,…,N)</td>
</tr>
<tr>
<td>$t_{prd}$</td>
<td>Time periods: Late Night (12am-4am), Early Morning (4am-8am), Morning (8am-12 pm), Mid-day (12 pm-4pm), Evening (4pm-8pm), Night (8pm-12am)</td>
</tr>
<tr>
<td>$q_t$</td>
<td>Traffic flow at time t</td>
</tr>
<tr>
<td>$\bar{q}_{td-1}$</td>
<td>Previous day mean traffic flow</td>
</tr>
<tr>
<td>$s_{qtd-1}$</td>
<td>Previous day standard deviation of traffic flow</td>
</tr>
<tr>
<td>$\bar{q}_{tprd-1}$</td>
<td>Previous time period mean traffic flow</td>
</tr>
<tr>
<td>$s_{qtd-1}$</td>
<td>Previous time period standard deviation of traffic flow</td>
</tr>
<tr>
<td>$S$</td>
<td>Mean speed over an hour</td>
</tr>
<tr>
<td>$T_l$</td>
<td>Hours left before landfall</td>
</tr>
<tr>
<td>$P_{evc}$</td>
<td>Cumulative population under mandatory evacuation</td>
</tr>
<tr>
<td>$d_{evc}$</td>
<td>Distance to the nearest evacuation zone from each detector</td>
</tr>
</tbody>
</table>

Evacuation demand data samples ($X^D_t$): We extract features related to evacuation traffic demand such as total population under mandatory evacuation zone, distance of a detector from the nearest evacuation zone, and hours left before hurricane landfall. In case of the variable “cumulative population under mandatory evacuation” we consider a time lag of 18 hr. between the declaration of evacuation order and the time when people start to evacuate. We perform an empirical analysis by running linear regression model multiple times with different time lag and
find that the coefficient associated with the variable total population under mandatory order is positive for 18-hour time lag and significantly influence the increase in evacuation traffic flow. The details on fixing the time lag is provided in [84]. Moreover, evacuation demand also depend on time period of the day; from our previous study [84] we found that people are more like to evacuate during day time compare to night time. Time periods are also considered to capture evacuation traffic demand.

Finally, the evacuation demand data samples have 9 features: Late Night (12am-4am), Early Morning (4am-8am), Morning (8am -12 am), Noon (12 pm-4pm), Evening(4pm-8pm), Night (8pm -12am), population under mandatory evacuation zone, distance of the nearest evacuation zone from each detector, hours left before hurricane landfall. We formulate the evacuation demand data as [number of samples (n), input time sequence (l), number of nodes (N), input features (c)] i.e. [120, 6, 806, 9].

**Graph Representation:** we follow several steps to construct the graph. Firstly, we map the detectors’ locations into Open Street map. Secondly, Considering the detectors as nodes we connect these detectors to complete the network. Finally, after constructing the network, we represent the network using adjacency matrix (see equation 5.1 and 5.2). We also calculate the travel distance between pairs of nodes from the open street map and estimate the travel time. We define the travel time between a pair of nodes as follows,

\[ t_{ij} = \frac{d_{ij}}{S_i + S_j} = \frac{2d_{ij}}{S_i + S_j} \]  \hspace{1cm} (5.10)

where, \( t_{ij}, d_{ij} \) indicate the travel time and distance between two consecutive detectors; \( S_i \) and \( S_j \) indicate average speed for two consecutive detectors. We use the travel time as weight for the adjacency matrix. We also perform gaussian transformation on the weighted graph adjacency matrix,
\[ A_t(i, j) = \begin{cases} \exp\left(\frac{-t_{ij}^2}{\sigma^2}\right), & \text{if } i \neq j \text{ and } \exp\left(\frac{-t_{ij}^2}{\sigma^2}\right) \geq r \\ 0, & \text{otherwise} \end{cases} \]  

where, \( \sigma \) indicate the standard deviation of travel time and \( r \) is the threshold to control the distribution and sparsity of weighted graph adjacency matrix, we fix the threshold value as 0.1 based on previous studies [32] and experiments results.

### 5.6.2 Baseline Models

We implement three baseline models to compare the performance of the proposed \( DGCN-LSTM \) model.

**LSTM:** In the LSTM model we use two stacked LSTM layer to prediction traffic for next 6 hour. Each of the layer we assign 4836 (number of nodes * output sequence length) hidden neurons. The output layer is a fully connected layer with a tanh activation function.

**Convolutional LSTM:** In the Convolutional LSTM (ConvLSTM) model, we stack a convolution layer with LSTM layer. Convolutional layer use convolution filter to extract the spatial correlation among traffic features in between consecutive detectors. We experiment with different size of the kernel \( k \) and find that the model performs best for a kernel size of 3. The output from the convolutional layer is fed into the LSTM layer to capture temporal correlation among traffic features while predicting traffic flow over a long sequence.

**Graph Convolutional LSTM:** In the graph convolutional LSTM (GCN-LSTM) model, we apply a similar approach as [26, 32]. In this case, the weights of the graph adjacency matrix are constant and assigned based on the distance between two consecutive nodes.
5.6.3 Model Training

We train the model using mean square error as the loss function. At each iteration, the model estimates the mean square error for the predicted flows \( \hat{F}_{t+p}^e \) and the actual flows \( F_{t+p}^e \) of the network. Afterward, the gradient of the loss function is backpropagated to adjust the weights to reduce loss function value. The loss function can be defined as,

\[
L = \text{Loss}(F_{t+p}^e, \hat{F}_{t+p}^e) \quad (5.12)
\]

\[
MSE = \frac{1}{p} \sum_{p=1}^p \frac{\sum_{E=1}(F_{t+p}^e - \hat{F}_{t+p}^e)^2}{E} \quad (5.13)
\]

where, \text{Loss}(\cdot) is the function to estimate the error between the actual \( F_{t+p}^e \) and estimated values \( \hat{F}_{t+p}^e \) and \( E \) denotes the set of links for the network. We implement our model using Pytorch environment [40] and train the model with dual NVIDIA Tesla V100 16GB PCIe GPU processor.

**DGCN-LSTM for Non-evacuation period:** From the regular traffic data samples, we use 90% for training, 5% for validation, and rest 5% of the data for testing the model. Based on the validation accuracy, we tune the hyperparameters such as learning rate, maximum number of iterations, and the type of the optimizer. We also track the training and validation loss values to check whether the model is overfitting or not. From the loss values, we find that it takes about 60 epochs with a learning rate of 0.001 for the model to converge (i.e., similar train and validation loss value). After that there are merely any variation in loss values (Figure 5.6 (a)). Moreover, after 70 epochs the value of the loss function for the validation data gradually starts increasing, indicating that the model starts to overfit. We use Adaptive Moment Estimation (ADAM) to train the model. Compared to other optimizers such as Adaptive Gradient (AdaGrad), Root Mean Square Propagation (RMSProp) etc., ADAM optimizer gives more stable solutions.

**Transfer Learned DGCN-LSTM for Evacuation period:** From the evacuation traffic data samples, we use 80% for training, 10% for validation, and rest 10% of the data for testing the
model. Similar to previous model, we experimented with different optimizer however ADAM optimizer gives the best result. We also track the changes in training and validation loss values to ensure the model is not overfitting. It takes 150 epochs for the model to converge after that it starts to overfits (Figure 5.6(b)).

Figure 5.6 Variations of training and validation loss (a) DGCN-LSTM (b) Transfer Learned DGCN-LSTM
5.7 Experiment results

Once the final model is fixed, we test it on the test data set. We calculate Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) as performance measures to check the accuracy of the implemented model. Performance metrics are defined as:

\[
RMSE = \sqrt{\frac{1}{p} \sum_{p=1}^{p} \sum_{E=1}^{E}(F_{t+p}^E - \hat{F}_{t+p}^E)^2}
\]  
\[
MAE = \frac{1}{p} \sum_{p=1}^{p} \sum_{E=1}^{E} |F_{t+p}^E - \hat{F}_{t+p}^E|
\]

In Table 5.2, we report the performance of the model on test dataset from regular period. To account the sensitivity of the model over different data samples, we randomly split the data to generate 10 different train, test, and validation data sets. Finally, we train 10 different models and report the mean and standard deviation of the estimated performance measures on the test data sets. Based on performance measures, we find that the proposed \textit{DGCN-LSTM} model performed best compared to other models. The RMSE and MAE values of the model are 226.85 and 133.82, respectively. However, RMSE and MAE provide aggregate information (average over all the outputs) on the performance of the models, hence, we also estimate the $R^2$ score. As shown in Table 5.2, the $R^2$ score for the proposed model is 0.98 which means the model can learn the regular traffic flow patterns very well.

However, when we apply the model for evacuation traffic prediction it performs poorly, the RMSE and MAE values significantly increased to 1440.994 and 1009.94, respectively. Hence, we use a transfer learning approach with additional demand features to capture the changes in traffic demand during hurricane period, which improves the overall prediction accuracy. For the transfer learned model the RMSE and MAE values are 399.69 and 268.03, respectively (Table 5.2).
Table 5.3 Comparisons among different models to predict traffic over 6-hour sequence

<table>
<thead>
<tr>
<th>Non-Evacuation Period</th>
<th>Mean RMSE</th>
<th>Std RMSE</th>
<th>Mean MAE</th>
<th>Std MAE</th>
<th>Mean R² Score</th>
<th>Std R² Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM</td>
<td>282.38</td>
<td>14.24</td>
<td>160.81</td>
<td>3.92</td>
<td>0.98</td>
<td>0.0028</td>
</tr>
<tr>
<td>GCN-LSTM</td>
<td>275.08</td>
<td>14.08</td>
<td>152.08</td>
<td>5.91</td>
<td>0.98</td>
<td>0.0024</td>
</tr>
<tr>
<td>ConvLSTM</td>
<td>246.57</td>
<td>31.19</td>
<td>146.11</td>
<td>14.94</td>
<td>0.98</td>
<td>0.0046</td>
</tr>
<tr>
<td>DGCN-LSTM</td>
<td><strong>226.84</strong></td>
<td>21.54</td>
<td><strong>133.82</strong></td>
<td>9.58</td>
<td><strong>0.98</strong></td>
<td>0.0032</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Evacuation Period</th>
<th>Min flow</th>
<th>Max flow</th>
<th>RMSE</th>
<th>MAE</th>
<th>R² Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>DGCN-LSTM</td>
<td>50.0</td>
<td>11,2470</td>
<td>1440.99</td>
<td>1009.94</td>
<td>0.21</td>
</tr>
<tr>
<td>Transfer learned DGCN-LSTM</td>
<td>50.0</td>
<td>11,2470</td>
<td>399.69</td>
<td>268.03</td>
<td>0.943</td>
</tr>
</tbody>
</table>

*Std = Standard Deviation*

Figure 5.7 shows the relation between actual and estimated link flows, indicating that the actual and predicted traffic flow almost match with each other. We also find that in some cases the model overestimates the flows. This is because of the drop in overall traffic at the upstream locations just before the landfall day due to changes in hurricane path and mandatory evacuation zones. We also observe the detector wise variations of actual and predicted traffic flows (Figure 5.8), which indicates that the model can capture spatiotemporal patterns of traffic very well.
Figure 5.7 Correlation between actual and predicted values for different prediction horizons

Figure 5.8 Actual and predicted value for different detectors for 6 hour prediction horizon
5.8 Congestion Mapping to Understand Network Disruption

We apply the implemented model to predict traffic for different days prior to hurricane landfall. As shown in Fig. 5.9, we map the predicted traffic flows to generate spatiotemporal traffic variations for different zones over a 6-hour period. The figure demonstrates congestion propagation at different zones of the network, such information will be critical for the emergency traffic management agencies to implement strategies focusing on reducing delay during hurricane evacuation. The figure provides further evidence that the implement model is able to capture the spatiotemporal traffic variations of a network during emergency evacuation even in case of unexpected event such as changes in hurricane causing zonal level evacuation traffic demand. For example, we observe a significant traffic congestion near west coast of Florida region: downstream of I75 and I4 on September 8 and 9, 2017. This is because of shift in hurricane path which forced people living in Naples, Cape Corals, Tampa, Levy, Jacksonville to evacuate at the eleventh hour causing significant congestion.
5.9 Conclusions

Evacuation traffic prediction is one of the most critical elements for activating pro-active traffic management strategies. However, evacuation traffic patterns differ from non-evacuation traffic condition such as higher traffic volume, no peaking behavior. Thus, it is more challenging to learn such irregularities using traditional network modeling approaches. Moreover, modeling spatiotemporal traffic variations requires large volume of data at a higher resolution, which is quite
impossible to get due to short span of evacuation period. By addressing these issues, this study develops a new method considering network dynamics to accurately predict evacuation traffic over multiple time steps.

First, we develop a deep learning architecture namely $DGCN$-$LSTM$ to learn the spatiotemporal network scale traffic patterns and train the model with non-evacuation period traffic data. Based on the experiment results, we find that the implemented $DGCN$-$LSTM$ outperforms the existing deep learning models such as $LSTM$, $ConvLSTM$ and $GCNLSTM$ model with an RMSE of 226.84. However, as we apply the model for evacuation period the RMSE value increased to 1440.99. We overcome this issue by adopting a transfer learning approach with additional evacuation traffic demand related features such as distance from the evacuation zone, time to landfall, and other zonal level features to control the information flow from the pretrained $DGCN$-$LSTM$ model. The final transfer learned $DGCN$-$LSTM$ model performs well to predict evacuation traffic flow (RMSE 399.69).
CHAPTER 6: CONCLUSIONS

This dissertation explored data-driven network modeling approaches to solve different transportation problems. Traditional network modeling approaches rely on many assumptions on user behavior and fail to account travel demand variations from real-time data. In case of high-fidelity network scale traffic prediction such approaches need to be modified to ingest high resolution data which in some cases are computationally expensive and require a substantial amount of time. Hence, these approaches are less suitable for real world high-fidelity traffic prediction such as intersection level traffic movement volume prediction. Moreover, these approaches are less robust to real time demand variations hence fail to capture traffic variations during emergency event such as hurricane evacuation. Considering these issues, we develop numerous task specific data-driven models for network wide traffic modeling and prediction. In summary, this study has focused on three objectives, the first objective is to develop a data-driven network scale model to solve the traffic assignment problem. The second objective is to develop a data-driven network model for intersection-level traffic movement volume prediction and the final objective is to investigate the scalability and transferability of such data-driven models for network-wide dynamic traffic prediction during evacuation period.

6.1 Summary of Major Results

This study provides key insights on performances of data-driven network models for large-scale traffic estimation and prediction. Such approaches will benefit proactive traffic control and management especially during emergency event such as hurricane evacuation. We have summarized the key findings of the study as follows:

- In the second chapter, we developed a data-driven approach based on a deep learning model to solve traffic assignment problems. We trained a Graph Convolutional Neural Network...
(GCNN) model to learn flow propagation from node to links for Sioux Falls Network and East Massachusetts Network. The validation result showed that GCNN model can capture the flow propagation of the network reasonably well, the mean absolute differences between the actual and estimated link flows are quite low for both Sioux Falls and East Massachusetts networks. The trained model can instantaneously predict the traffic flows, once the training is complete. Hence this approach can be used to overcome the challenges of deployment of mathematical programming or simulation-based traffic assignment models for large-scale networks. Furthermore, this method is completely data-driven without requiring any assumption on user behavior. Thus, it improves the reliability and stability of traffic assignment solutions. This method may open new opportunities to formulate alternative data-driven approaches for other network-oriented problems such as dynamic traffic assignment and network design problems.

• In the third chapter, we developed two deep learning architectures: Graph Convolutional LSTM (GCN-LSTM) and Graph Convolutional Encoder Decoder LSTM (GCN Encoder Decoder) models to predict intersection-level hourly traffic movement volumes over multiple time steps (e.g., 4-hour sequence). Such deep learning architectures capture the spatiotemporal cross correlation among network wide traffic features while learning the patterns in traffic movement volumes. To test the model performances, we have fused data from multiple sources such as travel demand data, built environment characteristics etc. We have extracted 1-year (2016) of traffic movement volume data from Seminole County’s automated signal performance measure (ATSPM) database. Experiment results show that the developed GCN-LSTM model outperforms all the other baseline models. Moreover, the absolute difference between actual and predicted volumes are quite low (GEH<5); for right turn, through and left turn movement RMSE values are 4.02, 59.37, and 2.47 respectively. The $R^2$ score for the model
is 0.98, which indicates that the model can capture the spatiotemporal variations of traffic movement volumes very well. Such a data-driven network-scale model is critical to ensure proactive decision making and optimal action plans for traffic operations and management.

- In the fourth chapter, we present an extensive analysis to understand the coverage and quality of a large number of detector data (recording traffic volume) for evacuation traffic analysis and modeling for Hurricane Irma. We conduct spatiotemporal data analysis to understand the changes in evacuation traffic pattern during Hurricane Irma. The analysis reveals that although evacuation order was placed early (September 6, 2017), a significant traffic congestion occurred at the downstream of I-75 and I-95 just before the landfall day. This happened because of the changes in hurricane path from the east coast to the west coast, forcing more people from south Florida and Jacksonville to evacuate at the last moment. The empirical analysis further reveals that there is at least about an 18-hr. time lag between the time of evacuation order and the time when people started to evacuate. Moreover, the location of the detectors with respect to evacuation zone, time left before hurricane landfall, and time of the days influence the variations of evacuation traffic. Such findings have potential implications in large network-scale evacuation traffic modeling.

- In the fifth chapter, we develop a deep learning architecture namely DGCN-LSTM to learn the spatiotemporal network scale traffic patterns and train the model with non-evacuation period traffic data. Based on the experiment results, we find that the implemented DGCN-LSTM outperforms the existing deep learning models such as LSTM, ConvLSTM, and GCNLSTM model with an RMSE of 226.84. However, as we apply the model for evacuation period the RMSE value increased to 1440.99. We overcome this issue by adopting a transfer learning approach with additional evacuation traffic demand related features such as distance from the
evacuation zone, time to landfall, and other zonal level features to control the information flow from the pretrained $DGCN$-$LSTM$ model. The final transfer learned $DGCN$$LSTM$ model performs well to predict evacuation traffic flow (RMSE 399.69). The implemented model can be applied to predict evacuation traffic over a longer forecasting horizon (6-hour). It can assist transportation agencies to activate appropriate traffic management strategies to reduce delays for evacuating traffic.

6.2 Limitations and Future Research Directions

This dissertation offers several contributions towards developing data-driven predictive analytics tools to support real-time traffic operations and management. However, it has few limitations which need to be addressed in future research.

- In the data-driven traffic assignment research, we provide empirical evidence that the proposed graph convolutional neural network performs well to capture user equilibrium traffic flow. However, we do not test our approach for real-world data due to the fact that existing GPS/vehicle trajectory-based user mobility data are not widely available to infer travel demand variations at a higher spatiotemporal resolution. Future research should explore alternative data augmentation approach to prepare travel demand data for such models.

- For the similar reason we do not use any information on real-time travel demand variations in case of traffic flow prediction models, rather we use hourly demand which is constant for all the days. High resolution demand data from emerging technologies such as mobile phone sensors or connected vehicles data can be used to overcome this issue. A comparison between traditional demand data and emerging data might give us more insights on model performance. However, currently the market penetration of such technologies is not representative enough to provide high-resolution travel demand information.
• Moreover, in case of evacuation traffic prediction we do not test our model for a different hurricane period. Mainly, due to lack of data from different hurricane evacuation periods. Future research can overcome this issue by integrating evacuation model with agent-based simulation environment (i.e., Anylogic) to test and re-train the model over different hurricane scenario.

• One of the limitations of graph convolutional neural network is that the model developed for one location is not transferable to other locations since the structure of the input graph is fixed. Future research should explore vectorizing the graph (graph to vector) to implement a spatially transferable model for networkwide traffic prediction.

• Another limitation is that, we cannot perform causality analysis using deep learning models, which means we cannot explicitly say, if a factor is positively or negatively influencing overall traffic condition. Future research should focus on interpretation of graph based deep learning models to understand causal relationship between inputs and target variables.

Despite some limitations this dissertation innovates data-driven methods to model network level traffic patterns and predict traffic flow variations using real-time data obtained from traffic sensors. Real-time network level traffic modeling will assist transportation agencies to understand the prevailing and future traffic conditions for each link of the network. Therefore, they will be able to make proactive decisions for traffic optimization and effective traffic management, especially in case of emergency events such as hurricane evacuation.
REFERENCES


2. Wardrop JG (1952) Some Theoretical. ROAD Eng Div Meet


and network control problem. 21st Int Teletraffic Congr ITC 21 Traffic Perform Issues Networks Futur - Final Program


